19991018 155

On the Cover

■ The ability to recognize and encode redundant patterns in data is critical to the success of a high-performance compression system. As a consequence, the choice of the data representation strategy used in a compression system can dramatically affect the performance of that system. To encode digital images with little or no perceptible loss of image quality and a reduced number of bits, a compression system must be capable of efficiently representing all possible images of interest without introducing artifacts that affect image utility.

The cover illustrates the real parts of a set of two-dimensional Gabor functions, which are formed from the product of a Gaussian function and complex exponential functions. These functions form the basis of a complete, nonseparable Gabor transform. Each Gabor basis function can be thought of as a spatial filter tuned to a specific band of spatial frequencies and orientations of local features. There is physiological evidence that animals and humans have similar filters in their brains, and that they utilize these filters for representing, or coding, visual images of their environment. Gabor basis functions form an effective representation system because they allow the creation of a visual model that minimizes the uncertainty of image information by simultaneously minimizing the uncertainty in the position, orientation, and spatial frequency of features in the images.

The article entitled "Synthetic Aperture Radar Image Coding," by Robert Baxter and Michael Seibert, describes how two-dimensional Gabor basis functions are used to encode images from synthetic aperture radars. The authors examine several methods of image coding by discussing image-coding concepts, image representation systems, compression-system design trade-offs, and system performance evaluation. The article focuses on the compression of detected synthetic-aperture-radar imagery for image-archiving applications in which the images are interpreted by image analysts.

Report Documentation Page

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERD **Journal Article**

4. TITLE AND SUBTITLE
SAR IMAGE CODING

5. FUNDING NUMBERS

6. AUTHOR(S)
BAXTER ROBERT A.

F19628-95-C-0002
PE-

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lincoln Laboratory, MIT
244 Wood Street
Lexington, MA 02173-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

JA-7535

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRES(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-98-073

11. SUPPLEMENTARY NOTES
LINCOLN LABORATORY
JOURNAL

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

■ Many factors govern the design of image-coding systems, ranging from the sensor type through storage and transmission limitations to the imagery end use. Image-compression performance depends on the costs of storage or transmission bandwidth (in bits per pixel), computational burden (in operations per second), information loss (utility), and artifacts that may change the receiver operating characteristics (probability of detection versus number of false alarms per area). We describe how general image-coding concepts are specialized for synthetic aperture radar (SAR) imagery, and we discuss design trade-offs with respect to representation systems, quantizers, and bit allocation. We also discuss criteria and techniques for evaluating the performance of compression systems.

14. SUBJECT TERMS

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT **Unclassified**	18. SECURITY CLASSIFICATION OF THIS PAGE **Unclassified**	19. SECURITY CLASSIFICATION OF ABSTRACT **Unclassified**	20. LIMITATION OF ABSTRACT **Unclassified**

Synthetic Aperture Radar Image Coding

Robert Baxter and Michael Seibert

■ Many factors govern the design of image-coding systems, ranging from the sensor type through storage and transmission limitations to the imagery end use. Image-compression performance depends on the costs of storage or transmission bandwidth (in bits per pixel), computational burden (in operations per second), information loss (utility), and artifacts that may change the receiver operating characteristics (probability of detection versus number of false alarms per area). We describe how general image-coding concepts are specialized for synthetic aperture radar (SAR) imagery, and we discuss design trade-offs with respect to representation systems, quantizers, and bit allocation. We also discuss criteria and techniques for evaluating the performance of compression systems.
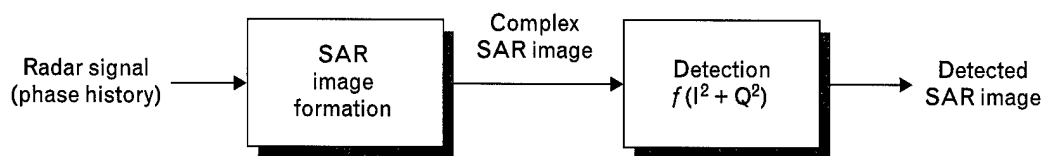
SYNTHETIC APERTURE RADAR (SAR) images formed from spatially overlapped radar phase histories are becoming increasingly important in a variety of remote sensing and tactical applications. The capability of SAR sensors to operate in virtually all types of weather conditions, from very long ranges and over wide areas of coverage, makes them extremely attractive for surveillance missions and for monitoring the earth's resources. With the increased popularity and corresponding abundance of such imagery, the need to compress SAR images without significant loss of image quality has become more urgent. In addition, because SAR images are interpreted for content by humans or by machines, appropriate coding of images enables efficient and effective machine selection and interpretation.

The need for compression of image data usually arises from two types of limitations: limited communications bandwidth or limited storage capacity. Figure 1 shows the SAR image-formation chain, with decreasing communications bandwidth requirements from left to right. The SAR image-formation stage involves the transformation of radar signals, both in-phase (I) and quadrature-phase (Q) components, via the Fourier transform and geometrical projections, to produce a complex-valued image. A *detected SAR im-* *age* is formed by sending the sum of the squares of the I and Q components through a nonlinear (typically logarithmic) transformation stage. Image analysts use the detected SAR image to interpret the SAR scene.

A compression system could be employed at three distinct points along the SAR image-formation chain: (1) compressing the radar signals prior to or in the process of forming the complex image, (2) compressing the complex image, or (3) compressing the detected image. Application-specific constraints and processing constraints ultimately determine which of these three options are selected. We have focused on applications that require human interpretation of SAR imagery, and, since humans interpret detected SAR images, our work on image compression has focused on the third option—compressing detected SAR images.

Figure 2 illustrates some of the differences between SAR imagery and natural imagery. Figure 2(a) shows a natural image of a rural scene containing a country store. (We use the term "natural image" to indicate that the image was not synthesized or computed and that it was derived from a sensor that operates in the visible part of the spectrum.) Most viewers can easily recognize familiar features such as roads, buildings, cars in the parking lot, trees in the fields, and perhaps

FIGURE 1. Synthetic aperture radar (SAR) image-formation chain. The complex SAR image is created from the Fourier transform and geometric projections of the radar signal. The detected SAR image, which is used by image analysts to interpret a SAR scene, is created by sending the sum of the squares of the in-phase (I) and quadrature-phase (Q) components of the complex SAR image through a nonlinear transformation stage.

even telephone poles along the road. Figure 2(b), a SAR image of the same rural scene shown in Figure 2(a), demonstrates how SAR images differ from natural images in several ways. First, the dynamic range of SAR images is typically much higher than natural images. To compensate for the increased dynamic range, it is common practice to use a preprocessing operation, such as taking the logarithm of image intensity values, to make the dynamic range more compatible with the human visual system. In addition, the spatial spectra of SAR images tend to have less spectral roll-off (higher spatial frequencies) than natural images. The high spatial-frequency content of SAR images is mainly caused by speckle noise and isolated point returns that represent localized scatterers or scattering

centers on electrically large objects. Objects can be identified in SAR images by the spatial arrangement and relative intensities of these isolated point returns. Radar shadows are also important in SAR images because they allude to the shape and material composition of objects in the scene.

Speckle noise in SAR imagery is multiplicative noise, whereas noise in natural images derived from optical sensors is mostly additive. Speckle noise tends to destroy the spatial redundancy that is common in natural images. Although a significant amount of research has been devoted to despeckling SAR images, it is not clear that despeckled imagery is easier to interpret or is preferred by image analysts.

Other differences between optical images and SAR
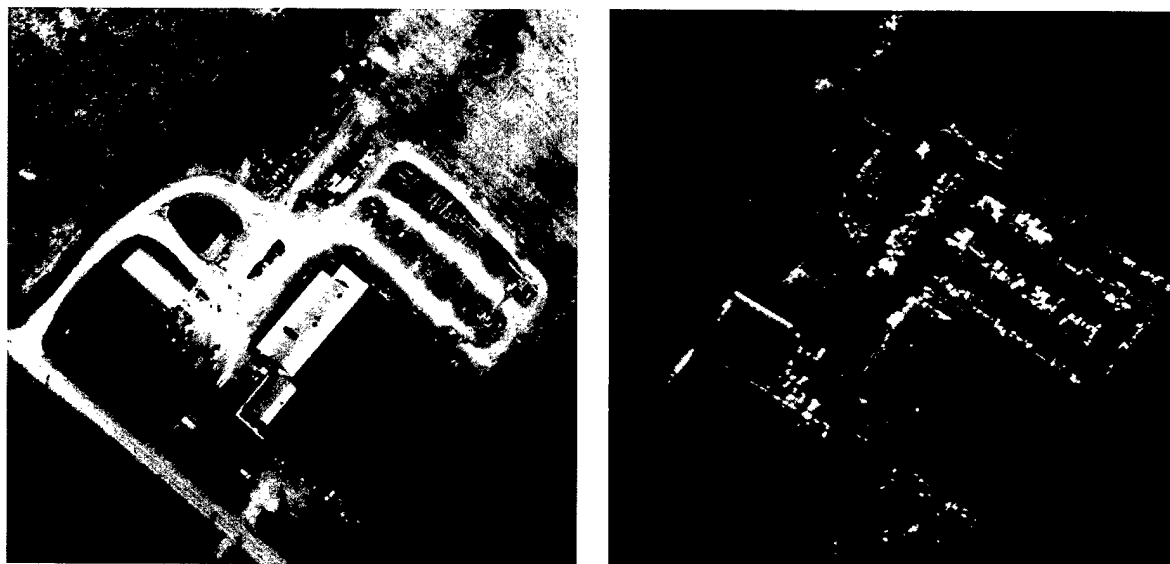
FIGURE 2. Images of a rural scene formed from (a) an optical sensor and (b) a SAR sensor. Familiar features such as roads, buildings, cars, and parking lots are easily recognizable in the optical scene. SAR images, however, have higher dynamic range, which requires special processing, and higher spatial-frequency content because of speckle noise and isolated point returns from localized scatterers. Objects in a SAR scene can be identified by the spatial arrangement and relative intensities of these isolated point returns.

images are related to the manner in which different objects in the scene project onto the two-dimensional image plane. For example, an effect known as *layover* in SAR imagery causes objects such as tall buildings to appear to lean toward the direction of illumination because the tops of these objects are illuminated first.

For reasons of economy and simplicity it is desirable to have one universal compression system for all types of images, but, as research has shown, compression systems designed for specific types of imagery tend to perform better than universal compression systems. Compression systems designed for specific applications, however, usually require more memory capacity and setup time because they utilize a predesigned compression codebook at the encoder and decoder stages.

Archiving of imagery is one of the most common applications for compression systems. It is also the most difficult application because the ultimate utilization of the imagery is often not known in advance. In the archiving application, the goal is to achieve as much compression as possible without significantly altering the image. Of course, the meaning of "significant" here depends on the possible applications of the imagery after archiving. Most of our work has focused on archiving applications in which there is a low tolerance for loss of image quality.

Background

The task of an image-compression system is to eliminate unnecessary and redundant information in the image and represent the remaining relevant information with the fewest number of bits. Consider the mathematical concept of the information contained in an image. An image is a collection of intensities located at distinct spatial locations, denoted by

$$I_k = I(x_k, y_k).$$

The set of possible intensities is denoted by

$$\{q_1, q_2, \ldots, q_L\},$$

which can be thought of as the set of allowable quantization levels.

Imagine that the image is to be sent over a communications channel so that we observe a stream of intensity values (for the moment, ignore the order in which the image is scanned). The probability that an intensity value is equal to intensity q_i is denoted by

$$p_i = P\{I(x, y) = q_i\}.$$

Information theory says that if we observe an intensity value with a low probability of occurrence, then it contains more information than an intensity value with a high probability of occurrence. In fact, if $p_i = 1$, information theory says that this intensity value carries no information. The presumed form of the information-content function is logarithmic, and if we want to measure information content in bits then the base 2 logarithm is used. In this way, information theory provides us with a mathematically precise definition of information.

Instead of the individual amounts of information in each intensity value, which could vary dramatically, we are more concerned with the average information content of the image. The average information content of an image can be computed as

$$H = -\sum_{i=1}^{L} p_i \log_2 p_i,$$

which is measured in bits per pixel (bpp). It turns out that H has a number of properties in common with thermodynamic entropy, so H is called the *entropy function*. The maximum value of H is obtained when the intensity values are equally distributed, i.e., when $H_{\max} = \log_2 L$. Thus, contrary to intuition, an image with a uniform distribution of intensity values contains the maximum amount of information— even if the image has no spatial correlations and looks like pure noise! The converse seems more intuitive: when all intensity values are the same, then $H = 0$, which implies that the image contains no information. For compression, lower values of H are more desirable.

Types of Redundancy

With a precise definition of information, we can attempt to be precise about redundancy. There are, however, at least three different types of redundancy in images. The first type is directly related to the entropy and we refer to it as *coding redundancy*, which is defined as
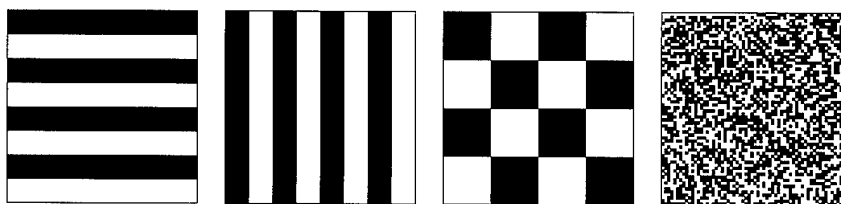
FIGURE 3. Four images with the same coding redundancy because the distribution of intensity values is identical, but with different spatial redundancies because of the different spatial correlations between pixels. Entropy coding methods that eliminate coding redundancy are not effective in eliminating spatial redundancy.

$$R_c = \frac{H_{max} - H}{H_{max}}.$$

If the distribution of intensity values in an image is uniform, the coding redundancy is zero; if all intensity values are the same, the coding redundancy is unity. Methods that attempt to eliminate coding redundancy are commonly called entropy coding methods. Many such methods exist; Huffman [1] and arithmetic coding [2] are among the most popular. Entropy coders are lossless because they eliminate redundancy without any loss of information.

A second type of redundancy is *spatial redundancy*. Figure 3 shows four figures that have the same coding redundancy ($R_c = 0$). Information theory says that each of these figures has the same amount of information ($H = 1$). Each of these figures, however, has a different type of spatial correlation between pixels. It is clear from the obvious visual differences in the four figures that entropy coding will not be of use in eliminating spatial redundancy. Fortunately, a number of other computational methods exist that do eliminate spatial redundancy to some degree. Transform-based coders attempt to eliminate spatial redundancy by changing the pixel-based image representation into another representation that is more efficient from a spatial-redundancy point of view. Run-length coding can also be used to eliminate spatial redundancy.

A third type of redundancy is *psychovisual redundancy*. Psychovisual, or perceptual, redundancy is a direct result of the properties of the human visual system (HVS). Consider the HVS as a black box. Images pass through this black box, and our perceptual representation of these images is the output. The properties of the HVS induce certain redundancies (of pos-

sibly different types) in the perceptual representation of the image. The HVS black box is highly nonlinear and depends on individual experience, attention, and mental state. We don't necessarily want to eliminate psychovisual redundancy; in fact, we often want to exploit it. For example, given that the HVS is less sensitive to high spatial frequencies than to low spatial frequencies (see the section entitled "Bit Allocation"), it tends to be easier to hide errors in the reconstructed image in regions with high spatial frequencies than in spatially smooth regions. As another example, consider the problem of detecting and classifying objects in a SAR image. On the basis of previous experience, we might expect to find certain symmetries in some objects, particularly at specific aspect angles. If the compression system destroys object symmetries or the spatial arrangement and relative intensities of isolated point returns, then this alteration of the image might lead to inappropriate object classification or even undetected objects.

Compression-System Components

Most compression systems can be decomposed into the components shown in Figure 4. The transform is an essentially lossless process that converts the image data at each pixel into a representation that tends to make it easier to eliminate spatial redundancy and hence is more appropriate and efficient for compression. In most compression systems, the transform produces sets of coefficients that we refer to as *coefficient subimages*. Given a desired total bit rate, the bit allocator determines how many bits should be allocated to each coefficient subimage. The bit allocator does not necessarily achieve the desired bit rate; the bit rate ultimately achieved is called the actual bit
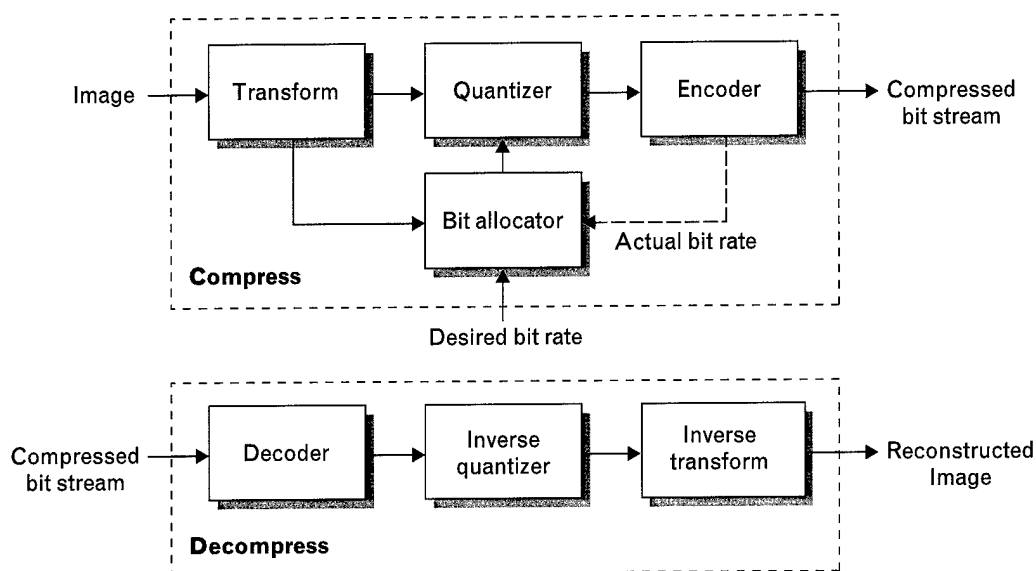
FIGURE 4. Compression-system and decompression-system components. The transform converts image data into coefficient subimages, which are converted by the quantizer into sets of discrete symbols at a bit rate determined by the bit allocator. The discrete symbols are then converted into a compressed bit stream by the encoder. To decompress an image the process is reversed.

rate. The quantizer, which is the major source of loss in image quality, converts the high-precision coefficients into sets of discrete symbols. These symbols are then converted into a compressed bit stream by the encoder. To decompress the image the process is reversed. The decoder converts the compressed bit stream into sets of symbols, which are dequantized into sets of coefficients, which are then inverse transformed to produce the reconstructed image.

Not shown in Figure 4 are the image preprocessor and the image postprocessor. The purpose of the preprocessor is to enhance features of interest in the image and attenuate irrelevant information, and the purpose of the postprocessor is to compensate for the degradation in image quality caused by the compression process. The type of preprocessor or postprocessor used is highly application dependent. If the application requires that the reconstructed image be as close as possible to the original image, then preprocessing and postprocessing may not be necessary. In our experience with SAR image compression, at bit rates below one bit per pixel, compression systems tend to change the histogram significantly enough that a perceptually more pleasing image is obtained with a postprocessor that simply remaps the histogram of the reconstructed image to that of the origi-

nal (which may require transmission or storage of side information in the compressed file). What actually happens is that the compression process decreases the contrast in the image, and the remapping process tends to restore the original tonal balance and contrast. For applications in which image enhancement is desirable, the preprocessor may serve to enhance certain image features and perhaps provide some type of noise reduction [3]. Most of the examples in this article assume the application requires that the reconstructed image match the original as closely as possible in a perceptual sense.

A variety of techniques can be chosen for each of the processing components shown in Figure 4. This degree of choice suggests a modular approach to compression-system design, which enables the system designer to choose components that yield the highest performance for a given set of constraints. As better techniques are discovered, the older compression-system components can be replaced with newer ones.

The JPEG Image-Compression Standard

The Joint Photographic Experts Group (JPEG) was formed in 1986 for the purpose of developing a universal standard for grayscale and color image compression. The word "Joint" indicates the cooperation

between the International Organization for Standardization, the International Telegraph and Telephone Consultative Committee, and the International Electrotechnical Commission. The name JPEG has now become associated with the standard itself. It is worth mentioning that JPEG compression was originally designed and optimized for natural imagery, and, because SAR imagery has different characteristics compared to natural imagery, we can expect that the performance of JPEG compression on SAR imagery may be somewhat deficient.

After researchers performed extensive subjective evaluation of several methods, the discrete cosine transform (DCT) was chosen in 1988 as the JPEG transform technique. In 1989, the first JPEG standard was adopted. In this first standard, and in the current JPEG standard, the DCT is applied to 8 × 8 disjoint subimages. The bit allocator is based on a quantization matrix, and the bit-allocation rate is controlled by a quality parameter set by the user, which has the effect of amplifying or attenuating the entries in the quantization matrix. The DCT coefficients are quantized by using the quantization matrix. Each number in the quantization matrix corresponds to a different two-dimensional spatial frequency, and represents the coefficient amplitude that is just detectable to the human eye. Dividing the DCT coefficients by their corresponding quantization-matrix elements and then rounding yields quantized coefficients, which are then entropy encoded into a bit stream. Either Huffman or adaptive arithmetic encoding can be employed.

The JPEG standard includes four operational modes: sequential lossy, progressive lossy, sequential lossless, and hierarchical. The sequential lossy and progressive lossy modes employ the DCT, whereas the sequential lossless mode uses a form of differential pulse-code modulation (DPCM) predictive coding. The hierarchical mode uses either DCT-based coding or DPCM predictive coding, and allows for progressive coding with increasing spatial resolution. Detailed descriptions of the JPEG standard can be found in G.K. Wallace [4] and W.B. Pennebaker and J.L. Mitchell [5].

The JPEG standard has been in widespread use since the early 1990s. At low bit rates, however, blocking artifacts in JPEG images can be visible and annoying. These blocking artifacts are the result of processing 8 × 8 blocks of the source image disjointly. At low bit rates, mismatches at block boundaries (due to the elimination of high-frequency coefficients) distort the overall appearance of the image.

A number of alternative methods of compression have been proposed to reduce or eliminate the blocking artifacts that characterize the JPEG standard at low bit rates. One of the most popular methods is to use the wavelet transform. The wavelet transform has three distinct advantages: it is inherently free of blocking artifacts, it naturally allows for progressive transmission, and it has a relatively low complexity. (The section entitled "Representation Systems" discusses the wavelet transform in more detail.) Currently, the JPEG standard is undergoing an evaluation process to determine how the present standard should be improved. The test images for evaluation have been derived from a variety of different sources. The baseline algorithm for the revised standard is based on a generalized wavelet-transform coder (specifically, the wavelet packet transform described in the next section).

Representation Systems

The ability to recognize and efficiently encode redundant patterns in data is a critical proficiency of any high-performance compression system. The choice of representation system used in a compression system can dramatically affect performance. For example, consider a sinusoidal signal. If we represent the signal as a set of amplitudes, our compression system must find an efficient way to compress these amplitude values. However, if our representation system transforms the amplitudes into a set of coefficients that represent amplitudes of various sine waves by using the Fourier transform, then most of the coefficients will be zero (in fact, for a pure sinusoid all coefficients would be zero except two). This example clearly shows that the choice of representation system can dramatically affect the compression performance. Of course SAR images are not so simple; nevertheless, the choice of representation can still significantly affect performance. The representation not only serves to eliminate spatial redundancy, but it also determines what

type of artifacts will be present in reconstructed images that are compressed at low bit rates.

Representational Strategies for a Signal Coding System

In designing a signal coding system we consider the following three representational strategies: (1) match the representation to the input signal, (2) match the representation to the task, or (3) match the representation to the user. The first strategy uses a representation system that efficiently encodes the characteristics of the input signal. For example, if the input signal consists of linear combinations of sine waves, then we would expect a representation system based on sine waves to efficiently encode the input signal. We would not expect the same sinusoidal representation system to efficiently encode a signal consisting of impulses and ramps. In general, the first representational strategy can be stated as the following optimization problem: find the smallest set of functions that when combined is complete with respect to the set of all possible images of interest (an additional constraint might also be imposed on the quantized coefficients such that the reconstructed image is as close as possible to the original image). Here, complete is meant to imply that the proper combination of functions reconstructs the image without error.

The second representational strategy attempts to find a representation system that makes the user more effective at performing the task at hand. For example, if the user is an algorithm that counts the number of objects of interest, then the representation system should make those objects easier to detect. Note that in this example it is not necessary to reconstruct the image without error; in fact, the reconstructed image need not look much like the original image. Clearly, this second representational strategy allows the possibility that the reconstructed image will be superior to the original image with respect to the user's performance of the task at hand.

If the goal is to make the reconstructed image match the original, this second representational strategy is inappropriate. Another problem with the second representational strategy is that, in practice, humans want to view the imagery and understand how an algorithm produces a given answer or conclusion. That is, in practice, it is almost always necessary to re-

construct imagery that looks similar to, if not perceptually identical to, the original imagery. Otherwise, the original imagery is lost forever.

The third representational strategy attempts to find a representation system similar to that of the user. Suppose we are able to construct a representation system that mimics that of the human visual system (HVS), and we assume that a change in the output of this HVS model corresponds to a perceived change in the image. Given such a model, it is possible to control, minimize, and manipulate perceived differences between an original image and a compressed and reconstructed image. We call this approach HVS-guided compression.

Characteristics of the HVS Representation

Constructing a model of the HVS representation is difficult because the HVS is a complex nonlinear system. Over the past few decades, however, an enormous amount of information about how the brain processes images has been gathered by neuroscientists and psychophysicists. We now know that the HVS representation consists of filters that cover spatially overlapping regions of the visual field. These filters, which come in multiple spatial scales and orientations, collectively form a redundant nonorthogonal representation of visual scenes.

Cortical Filters. The actual filters used by the HVS cannot be measured directly, but the filters used by the cat's visual system and those of other animals have been empirically determined. These filters are measured by inserting a small probe into an anesthetized animal's visual cortex near a neuron and recording the response of this neuron to a small spot of light. The firing rate of the neuron as a function of the location of the light is referred to as the receptive-field profile of the neuron. This receptive-field profile can be thought of as a filter. The top row of Figure 5 shows the measured filter characteristics of three different neurons in a cat's visual cortex (Area V1). It turns out that two-dimensional Gabor functions, which are Gaussian functions modulated by sinusoids, fit the receptive-field profile data quite well. The middle row in Figure 5 shows the best-fitting two-dimensional Gabor functions for each of the neurons, and the bottom row shows that the (Chi-square) residual error
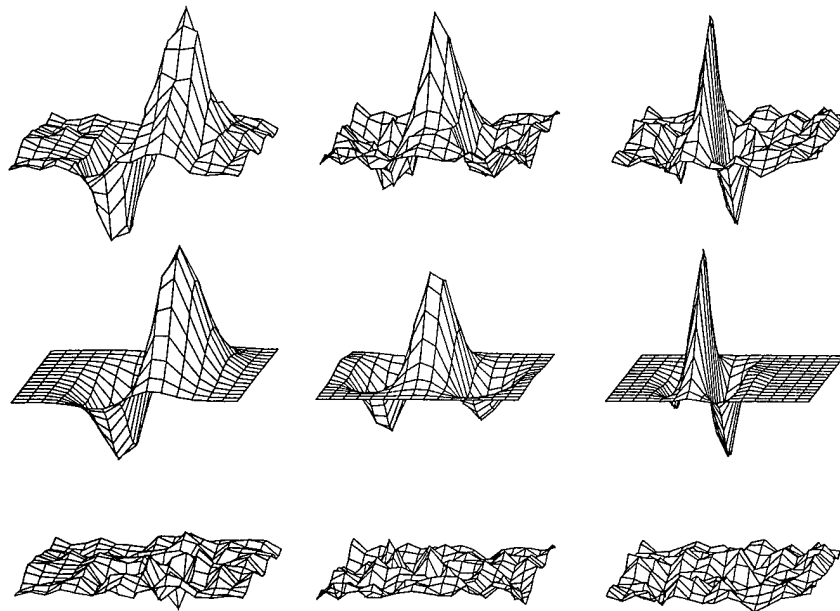
FIGURE 5. Comparison of three measured receptive-field profiles of cortical neurons in a cat's visual system and two-dimensional Gabor functions that model these receptive fields. The top row shows the measured receptive-field data and the middle row shows the best-fitting two-dimensional Gabor functions. The bottom row shows the Chi-squared residual error between the measured data and the theoretical data. (Reprinted with permission from J.G. Daugman [8].)

between the Gabor functions and the measured profiles is negligible. Note that these cortical filters are tuned to different spatial frequencies (compare the first and third images in each row), they can be odd or even functions (compare the first and second images), and they can have different polarities (compare the first and third images).

While these cortical filters clearly play a part in the HVS representation, it is not clear how they combine to form the perception of a visual scene. From a mathematical point of view, a linear combination of two-dimensional Gabor functions of different spatial scales, locations, and orientations can be constructed such that any image can be represented. Such a representation is redundant and nonorthogonal—like the HVS representation. We refer to this transform as an HVS-like Gabor wavelet transform. Although the nonorthogonality complicates the reconstruction process somewhat, near perfect reconstruction is possible [6]. The redundancy of the HVS representation seems to be inconsistent with the compression objective because many more filter coefficients are needed than the number of pixels in the image. (The redun-

dancy of the HVS representation may be consistent with a broader set of objectives related to the survivability of the human species.) Even if we retain only the significant coefficients, the process of deciding which coefficients to keep tends to be computationally intensive—especially when this representation is implemented on a single-processor computer.

The computational burden is much less if we consider reducing the number of spatial scales or reducing the number of orientations. Two alternative image transforms are the two-dimensional Gabor transform and the separable two-dimensional wavelet transform. The filters, or basis functions, used in the two-dimensional Gabor transform are quite orientation selective, but they cover only a single spatial scale. The wavelet transform covers multiple spatial scales, but the separable two-dimensional wavelet transform has poor orientational selectivity due to its separable construction.

Gabor Transform

The Gabor transform is a windowed Fourier transform with a Gaussian window function. The one-

dimensional Gabor transform was invented in 1946 by D. Gabor [7], and J.G. Daugman showed how it could be extended to two dimensions [8]. A Gabor function is a Gaussian modulated by a complex exponential. In two dimensions, a Gabor elementary function (GEF) has the form

$$g_{pqrs}(x, y) = e^{-[(x-p)^2+\beta(y-q)^2]/\alpha^2} e^{i(\Omega_r xr + \Omega_s ys)}$$
$$= g(x - p, y - q)e^{i(\Omega_r xr + \Omega_s ys)} , \qquad (1)$$

where the term $g(\cdot)$ represents a two-dimensional Gaussian window function centered at the spatial location (p, q). The parameter α controls the spatial extent of the window. The parameter β controls the eccentricity of the two-dimensional Gaussian shape; here we assume $\beta = 1$. The GEF g_{pqrs} is tuned to spatial frequency $(\Omega_r r, \Omega_s s)$; (r, s) are the harmonic numbers corresponding to this spatial frequency.

The two-dimensional Gabor transform is a combined spatial-spectral transform that represents an image as a set of spatially overlapping two-dimensional Gabor functions. The Gabor expansion of an image $I(x, y)$ can be expressed as

$$I(x, y) = \sum_{p,q} \sum_{r,s} c_{pqrs} g_{pqrs}(x, y) .$$

Note that the Gabor transform linearly samples spatial frequencies.

If there are fewer coefficients c_{pqrs} than the number of samples (the undersampled case), we cannot guarantee that this decomposition will represent all possible images. The fewest number of coefficients required to represent all possible images is equal to the number of samples (pixels) in the image. This is called the critically sampled case, or the complete case. For the Gabor transform to have the same number of coefficients as pixels, we constrain the number of spatial samples $(n_p \times n_q)$ and the number of spatial-frequency samples $(n_r \times n_s)$ by

$$N = WH = n_p n_q n_r n_s ,$$

where N is the number of pixels in the image, and W and H are the pixel width and height of the image.

The implication of this constraint on the sampling intervals can be most easily understood by assuming for the moment that the image is square. In this case

we have the constraint

$$N = W^2 = n_p^2 n_r^2 .$$

This constraint implies that the spatial sampling interval is equal to the number of spatial-frequency samples; i.e., $\Delta_x = n_r$ since $\Delta_x \equiv W/n_p$. Generalizing this result to non-square images yields the following restrictions on the spatial sampling intervals:

$$\Delta_x = n_r, \quad \Delta_y = n_s .$$

The fundamental spatial frequencies Ω_r and Ω_s are related to the spatial sampling intervals by

$$\Delta_x \Omega_r = \Delta_y \Omega_s = 2\pi .$$

With these constraints, the complete Gabor expansion becomes

$$I(x, y) = \sum_{p=0}^{n_p-1} \sum_{q=0}^{n_q-1} \sum_{r=0}^{n_r-1} \sum_{s=0}^{n_s-1} c_{pqrs}$$
$$\times e^{-[(x-p)^2+(y-q)^2]/\alpha^2} e^{-i2\pi(xr/n_r + ys/n_s)} . \qquad (2)$$

Visualizing the Gabor Basis Functions. The set of functions $g_{pqrs}(x, y)$ are the complex basis functions for the complete Gabor transform. Figure 6 shows these basis functions for $n_r = n_s = 8$ (64 basis functions), and Figure 7 shows how the frequency components are arranged in Figure 6. At each spatial sampling position in a given image, with these parameters there are $n_r n_s = 64$ basis functions. For a 512×256-pixel image there are 64×32 overlapping Gabor neighborhoods. The coefficients of these basis functions provide a local frequency analysis of the image in each neighborhood. The spatial extent of each neighborhood, as well as the degree to which the basis functions overlap, is controlled by α (see Equation 1). The orientation-selective properties of the Gabor basis functions are apparent in Figure 6, which illustrates how orientational selectivity increases as radial frequency increases.

Computing Gabor Coefficients with the Zak Transform. Since the Gabor transform is nonorthogonal, computation of the coefficients is more complicated than filtering with each basis function and downsampling. At least three different methods of computing Gabor coefficients have been proposed. M. Bas-

(a)   (b)
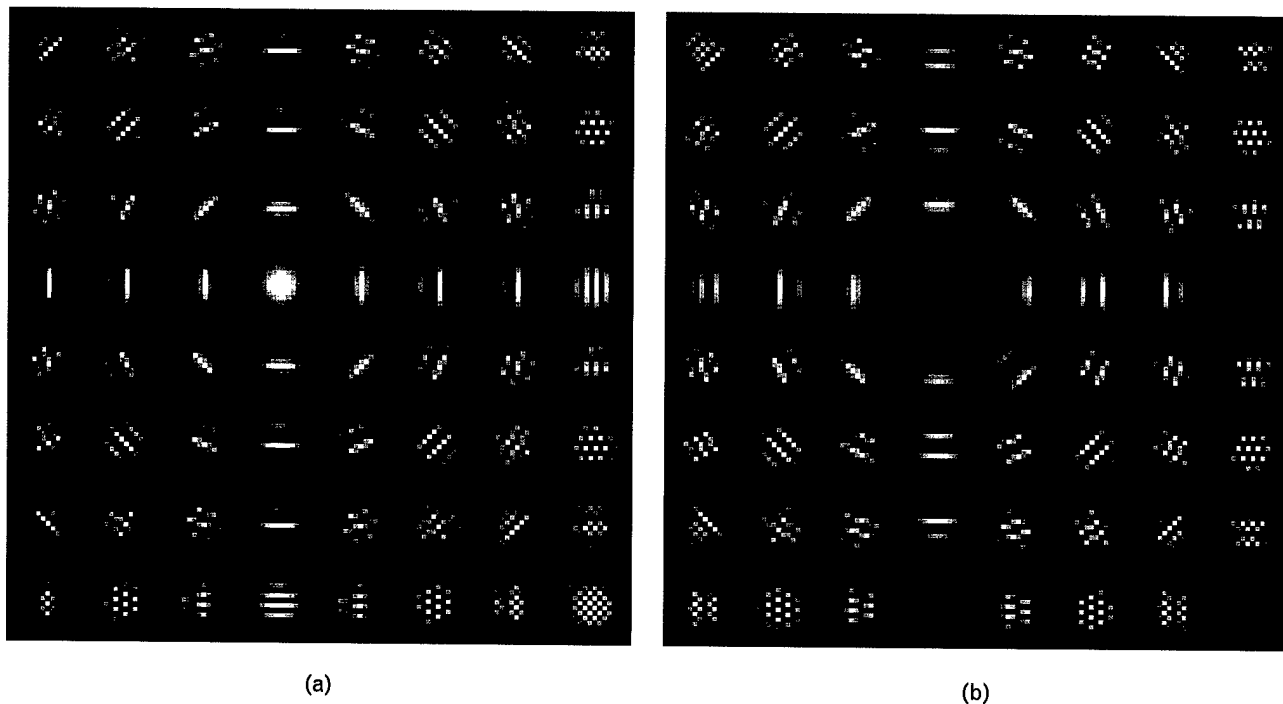
**FIGURE 6.** (a) Real components and (b) imaginary components of the Gabor-transform complex basis functions for $n_r = n_s = 8$. The two-dimensional Gabor transform represents a given image as a set of spatially overlapping two-dimensional Gabor functions. At each spatial sampling position in an image there are $n_r n_s$ basis functions of limited spatial extent that form a Gabor neighborhood. The coefficients of these basis functions provide a local frequency analysis of the image in each neighborhood.
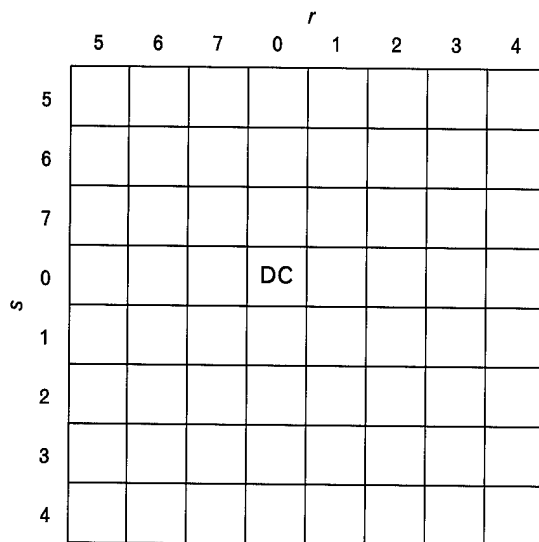
FIGURE 7. Arrangement of the Gabor-transform basis functions shown in Figure 6. The basis function associated with the unmodulated DC component ($r = s = 0$) is simply the (shifted) Gaussian window function. The remaining basis functions are modulated at spatial frequencies related to (r, s). Indices of 5, 6, and 7 correspond to negative spatial frequencies.

tiaans described a biorthogonal expansion method in which an auxiliary window function is made biorthogonal to the Gaussian window [9]. Daugman described a gradient descent method in which the coefficients are viewed as weights in an adaptive network [10]. The gradient descent procedure adjusts the weights such that the squared difference between the image and the value of the Gabor expansion is minimized across all pixels. L. Auslander et al. [11] as well as I. Gertner and Y.Y. Zeevi [12] showed how the Zak transform could be used to determine Gabor coefficients (this method is sometimes referred to as the Zak-Gabor transform). Because of the numerical efficiency of this approach, the Zak-Gabor transform is preferred.

The Zak transform can be used to compute the Gabor coefficients c_{pqrs} as follows. The two-dimensional discrete Zak transform computes the Fourier transform of a spatially decimated image (subsampled by n_p and n_q in each spatial dimension) offset by (p, q). The discrete Zak transform of I is given by
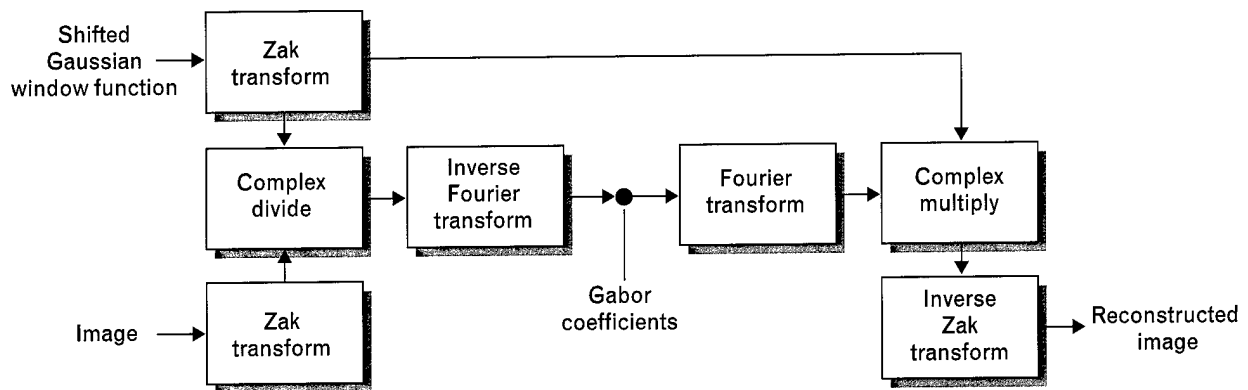
FIGURE 8. Processing steps in computing the Zak-Gabor transform. From Equation 3, the Gabor coefficients are the Fourier coefficients of the ratio of the Zak transforms of the image and the shifted Gaussian window function. We assume the image is periodically extended in each dimension so that the fast Fourier transform (FFT) can be used to compute the Gabor coefficients and reconstruct the image. The Zak transforms require two-dimensional FFTs, while the Fourier transforms require four-dimensional FFTs.

$$Z(I)(p, q, \rho, \sigma) =$$

$$\sum_{r=0}^{n_r-1} \sum_{s=0}^{n_s-1} I(p + rn_p, q + sn_q)e^{-i2\pi(rp/n_r + s\sigma/n_s)} ,$$

where the transform is evaluated at spatial coordinates (p, q) and at spatial-frequency coordinates (ρ, σ).

Taking the discrete Zak transform of both sides of Equation 2 yields

$$Z(I)(p, q, \rho, \sigma) =$$

$$Z(g)(p, q, \rho, \sigma) \sum_{k=0}^{n_p-1} \sum_{l=0}^{n_q-1} \sum_{m=0}^{n_r-1} \sum_{n=0}^{n_s-1} \left\{ c_{klmn} \right.$$

$$\left. \times e^{i2\pi(pk/n_r + ql/n_s)} e^{i2\pi(pm/n_r + \sigma n/n_s)} \right\}.$$

With this expression, the Gabor coefficients can be computed as

$$c_{pqrs} = \sum_{m=0}^{n_p-1} \sum_{n=0}^{n_q-1} \sum_{\rho=0}^{n_r-1} \sum_{\sigma=0}^{n_s-1} \left\{ \frac{Z(I)(m, n, \rho, \sigma)}{Z(g)(m, n, \rho, \sigma)} \right.$$

$$\left. \times e^{-i2\pi(mp/n_r + nq/n_s)} e^{-i2\pi(\rho r/n_p + \sigma s/n_q)} \right\}$$

(3)

$$= DFT\left\{ \frac{Z(I)}{Z(g)} \right\},$$

where $DFT\{\cdot\}$ denotes a discrete (four-dimensional) Fourier transform.

Numerical Implementation Using the Fast Fourier Transform. Equation 3 shows that the Gabor coefficients are the Fourier coefficients of the ratio of the Zak transforms of the image and the Gaussian window function. If we assume that the image is periodic in each dimension, then we can use the fast Fourier transform (FFT) to compute the coefficients efficiently. Figure 8 illustrates the processing steps in computing the Gabor coefficients and reconstructing the image. Note that in Figure 8 the Zak transforms require two-dimensional FFTs, whereas the Fourier transforms require four-dimensional FFTs.

Modification for Numerical Stability. Equation 3 reveals that it is crucial that $Z(g)$ be nonzero. In fact, for a Gaussian window, $Z(g) = 0$. Therefore, in a strict sense, the Zak-Gabor transform does not converge. However, K. Assaleh, Zeevi, and Gertner have shown that if the Gaussian window is slightly shifted, its Zak transform remains finite and the transform is numerically stable [13]. Therefore, we shift the Gaussian window by one-half pixel in each dimension; i.e., the window function becomes

$$g(x - p, y - q) = e^{-[(x-p+1/2)^2 + (y-q+1/2)^2]/\alpha^2} .$$

Visualizing Gabor Coefficients. There are several ways of visualizing Gabor coefficients. An intuitive understanding of the coefficients can be gained from viewing the set of coefficients associated with each frequency component of the transform as a filtered
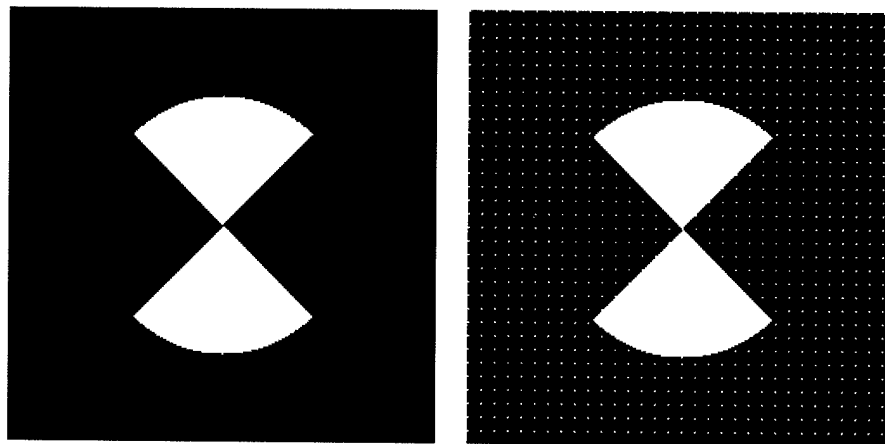
FIGURE 9. (a) 256 × 256-pixel test image; (b) test image with a 32 × 32 spatial sampling grid (white dots) superimposed. The spatial sampling grid identifies the centers of all Gabor neighborhoods in the image.

and subsampled version of the image (with corrections to account for the nonorthogonality of the transform). Of course, since the coefficients are complex numbers, each of these subimages is complex. Thus it is helpful to view them as two subimages—one that represents the coefficient magnitudes and another that represents the coefficient phases.

Figure 9 shows a 256 × 256-pixel test image with a

32 × 32 spatial sampling grid, and Figure 10 shows the corresponding Gabor coefficients of the test image. The spatial sampling grid identifies the centers of all Gabor neighborhoods in the image, i.e., all spatial locations (p, q). The coefficients associated with the frequency harmonic corresponding to $(r, s) = (0, 0)$ are referred to as the DC component because they are not modulated; coefficients associated with the other

(a)   (b)
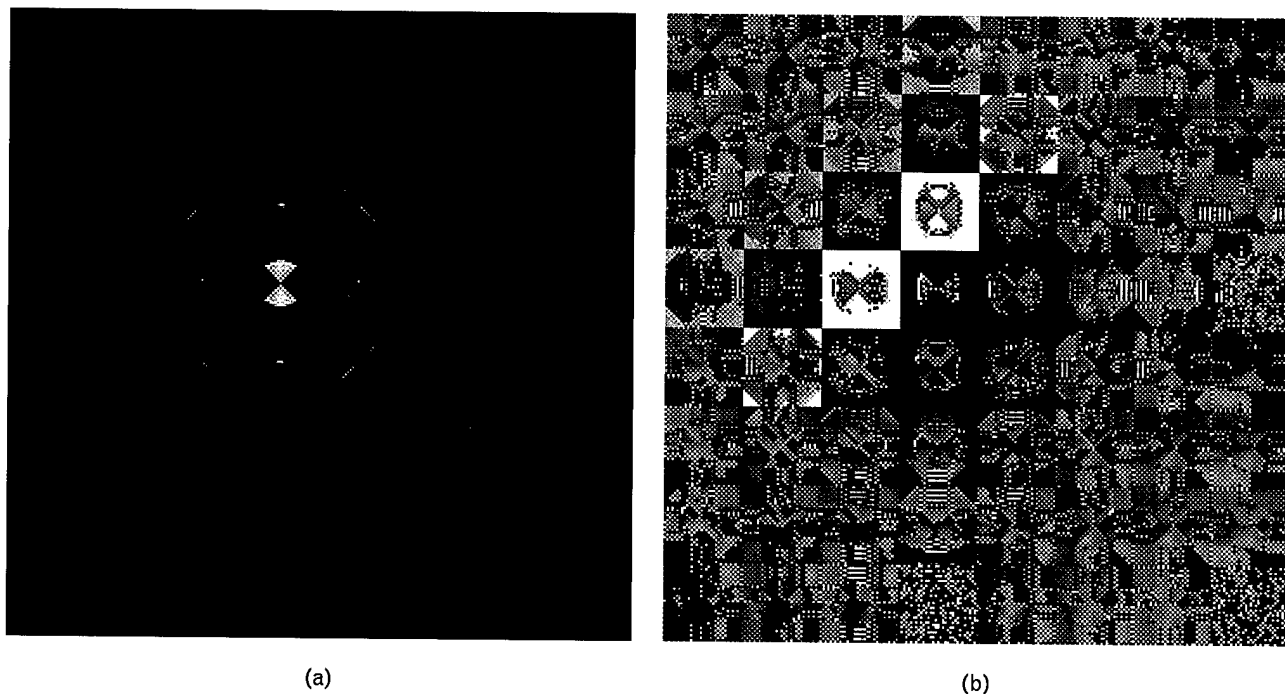
FIGURE 10. Gabor coefficients corresponding to the test image in Figure 9: (a) coefficient magnitudes and (b) coefficient phases. The subimages can be thought of as filtered and subsampled versions of the test image.

frequency harmonics are referred to as the AC component. The basis function associated with the DC component is simply the (shifted) Gaussian window function. Because this basis function is real, and because the image is real, the coefficients associated with the DC component are real. Similarly, coefficients associated with the $(n_r/2, 0)$, $(0, n_s/2)$, and $(n_r/2, n_s/2)$ frequency components are also real. For real images, the remaining coefficients exhibit conjugate symmetry about the origin of the spatial-frequency plane. Note that, because of conjugate symmetry, the number of values required to uniquely specify the Gabor coefficients is exactly equal to the number of pixels in the image.

As mentioned previously, the subimages in Figure 10 can be thought of as filtered and subsampled versions of the image with corrections to account for the nonorthogonality of the basis functions. Specifically, with $n_r = n_s = 8$ and $n_p = n_q = 32$, these coefficient subimages are effectively downsampled by a factor of eight to produce 32×32-pixel images. This downsampled image is easily recognized in the magnitude

subimage corresponding to the DC component, which resembles a low-pass filtered and downsampled version of the image. The artifacts present in this subimage are caused by the corrections for the nonorthogonal basis functions and by the periodic boundary conditions that are enforced by the FFT.

Wavelet Transform

As the previous subsection explained, the Gabor transform uses a fixed window size, and Gabor basis functions sample frequency space linearly in each dimension. A wavelet is a waveform of limited spatial extent that has an average value of zero. Wavelet basis functions use larger window sizes at lower frequencies and smaller window sizes at higher frequencies. Because the spatial extent of the window changes, wavelets analyze signals on multiple spatial scales. The wavelet transform is essentially a decomposition onto a set of frequency bands of approximately equal width on a logarithmic scale. Therefore, we characterize the Gabor and wavelet transforms as multifrequency and multiscale representations, respectively. Figure 11
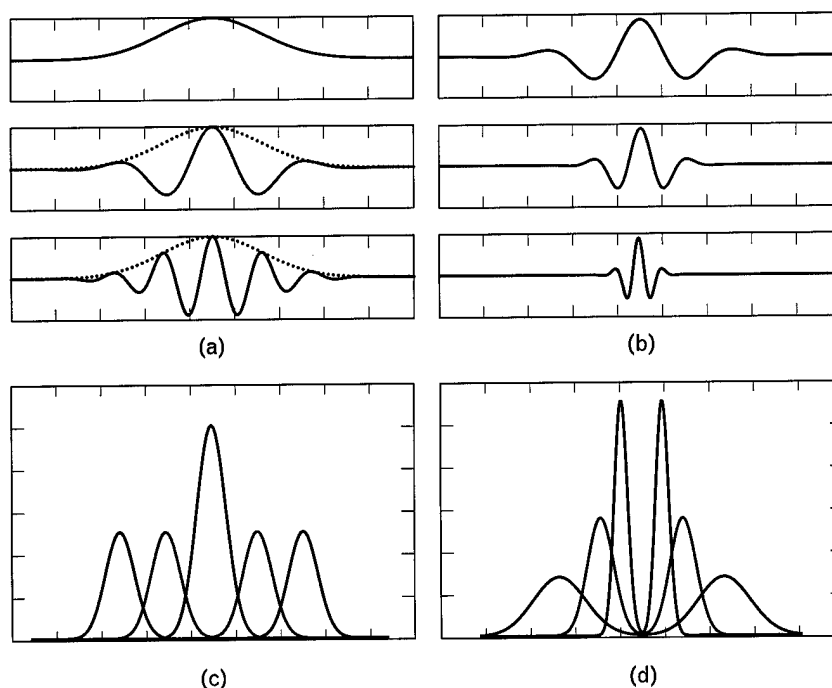
FIGURE 11. Comparison of Gabor basis functions and wavelet basis functions in one dimension. (a) Gabor basis functions in one-dimensional spatial domain, (b) wavelet basis functions in one-dimensional spatial domain, (c) Gabor basis functions in one-dimensional spatial-frequency domain, and (d) wavelet basis functions in one-dimensional spatial-frequency domain.

Decomposition   **Reconstruction**
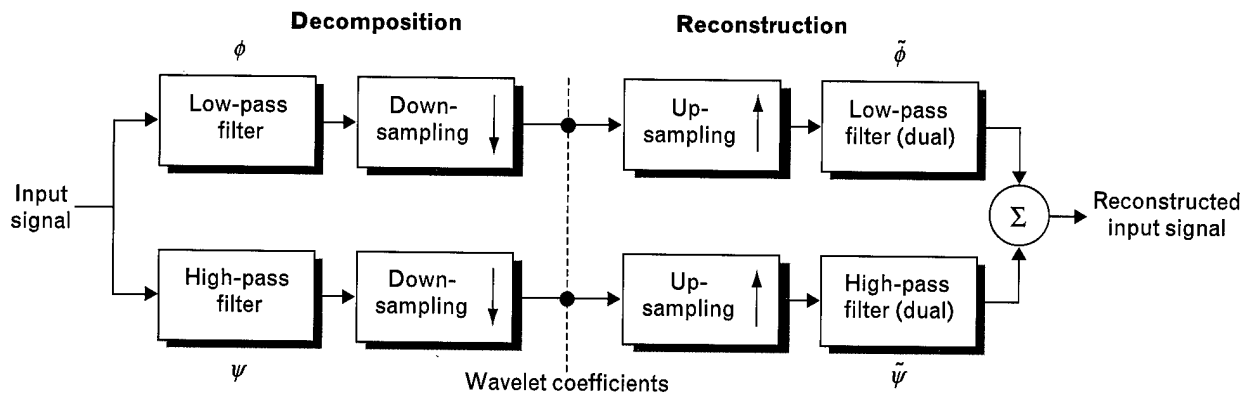
FIGURE 12. Processing steps in a single-level one-dimensional wavelet transform. Decomposition consists of filtering and downsampling, and reconstruction involves upsampling and filtering. The input signal is decomposed into a set of wavelet coefficients and then reconstructed to yield a nearly perfect estimate of the input signal. The low-pass filter and the high-pass filter, which are associated with the scaling function ϕ and the wavelet function ψ, respectively, obey specific constraints such that they reconstruct the signal with negligible error.

compares Gabor and wavelet basis functions in one dimension, both in the spatial and spatial-frequency domains. For a detailed comparison of wavelet and Gabor representations see S.G. Mallat [14].

The wavelet transform actually involves two functions: the wavelet function, which is denoted by ψ, and the scaling function, which is denoted by ϕ. Unlike the wavelet function, the scaling function has a nonzero average value. In terms of quadrature mirror filters, the wavelet function is determined by the high-pass filter and the scaling function is determined by the low-pass filter.

Wavelet functions are self-similar; i.e., they can be generated from a single function by translation and scaling. Denoting the wavelet function in one dimension by ψ, we can construct a family of wavelets

$$\psi^{(a,b)}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right)$$

that are scaled by a and translated by b. For certain choices of wavelet and scaling functions, perfect reconstruction is possible with a discrete wavelet transform if we restrict the values of a and b to a discrete grid. The range of choices significantly increases if we use a biorthogonal transform, i.e., if the scaling and wavelet functions for decomposition and reconstruction are different. In this case, the decomposition and reconstruction filters are called dual filters; for example, ψ and $\tilde{\psi}$ are dual filters. If the same scaling

and wavelet functions are used for decomposition and reconstruction, and reconstruction is perfect, the transform is orthogonal.

Figure 12 denotes the signal processing steps involved in a single-level wavelet decomposition. Decomposition consists of filtering and downsampling, and reconstruction involves upsampling and filtering. The low-pass and high-pass filters obey specific constraints such that they reconstruct the signal with negligible error. For orthogonal wavelets, the filters and their corresponding dual filters are identical; for the more general case of biorthogonal wavelets they are not. Figure 13 depicts multiple-level wavelet decomposition as a binary tree. At each level, low-pass and high-pass components are generated and the low-
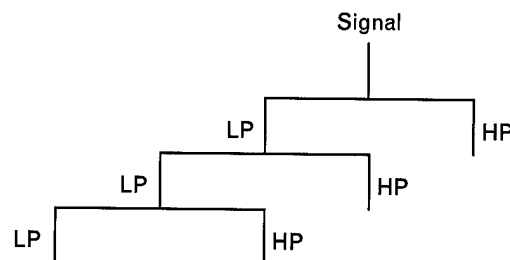
FIGURE 13. Binary-tree diagram of a multiple-level one-dimensional wavelet transform. Branches labeled LP are low-pass filters and branches labeled HP are high-pass filters. At each level, low-pass and high-pass components are generated and the low-pass component of the previous level is used as the input to the next decomposition stage.

pass component of the previous level is used as the input to the next decomposition stage. In the literature, the low-pass component is often called the approximation signal and the high-pass component is called the detail signal [15].
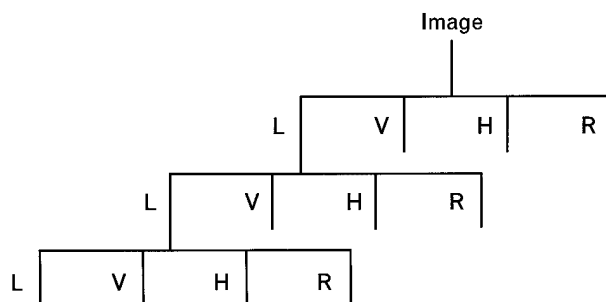
The separable two-dimensional wavelet transform can be constructed from the one-dimensional wavelet transform by combining horizontally oriented wavelets and vertically oriented wavelets [15]. Two-dimensional wavelets are defined as a tensor product of the one-dimensional wavelets. The two-dimensional (unoriented) scaling function is the product of two one-dimensional scaling functions:
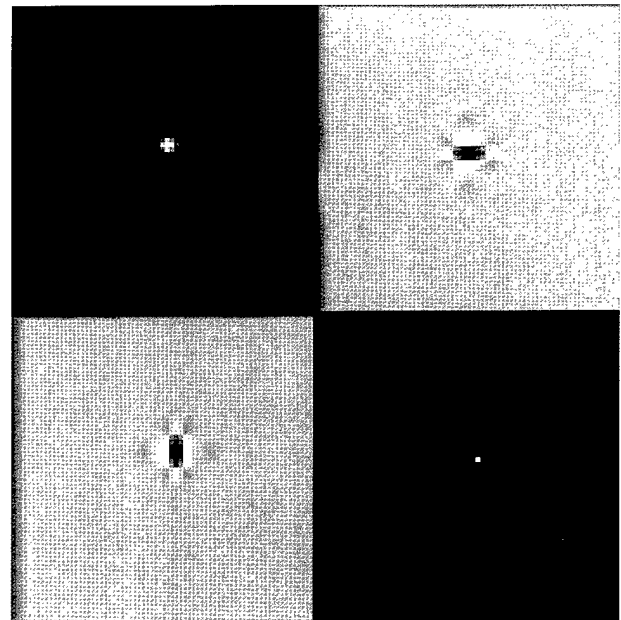
$$\phi(x, y) = \phi(x)\phi(y).$$

The two-dimensional wavelets are

$$\psi_V(x, y) = \phi(x)\psi(y)$$
$$\psi_H(x, y) = \psi(x)\phi(y)$$
$$\psi_R(x, y) = \psi(x)\psi(y),$$

where the subscripts stand for vertical, horizontal, and high-pass residual components. Figure 14 depicts the multiple-level wavelet decomposition in two dimensions as a quadtree. At each level, low-pass (L), vertical (V), horizontal (H), and high-pass residual (R) components are generated. The low-pass component from the previous level is used as the input signal to the next level. Figure 15 shows the separable wavelet basis functions corresponding to a single spatial scale. Here we have used seven and nine tap biorthogonal filters (i.e., the decomposition wavelet fil-

FIGURE 15. Separable-wavelet-transform basis functions corresponding to a single spatial scale.

ter consists of seven coefficients and the reconstruction filter consists of nine coefficients or vice versa), as described in M. Antonini et al. [16]. Figure 16 shows a three-level separable wavelet decomposition of the test image shown in Figure 9. For each level, the upper left subimage corresponds to the low-pass coefficients, the upper right subimage corresponds to the vertical coefficients, the lower left subimage corresponds to the horizontal coefficients, and the lower right subimage corresponds to the high-pass residual coefficients.

Wavelet Packet Transform

The wavelet packet transform is a generalization of the wavelet transform that provides a richer range of possibilities for signal analysis. Instead of using only low-pass components as inputs to the next level, the wavelet packet decomposition considers the full tree and chooses among the many possible representations available. Figure 17(a) depicts the separable two-dimensional full wavelet packet tree, and Figures 17(b), (c), and (d) depict three different decompositions, including a Gabor-like representation in Figure 17(c) and a wavelet representation in Figure 17(b). To emphasize the difference between the general wavelet packet decomposition and the more restrictive wave-

FIGURE 14. Quadtree diagram of a multiple-level two-dimensional separable wavelet transform. Branches labeled L represent low-pass filters, branches labeled V represent vertical filters, branches labeled H represent horizontal filters, and branches labeled R represent high-pass residual filters.
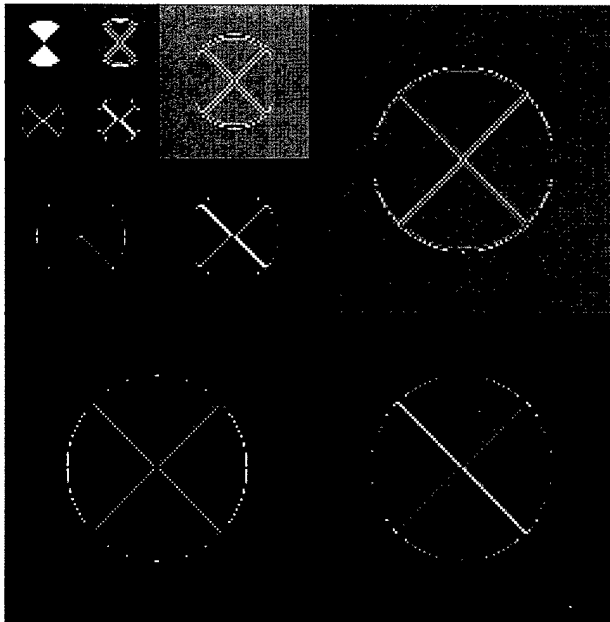
FIGURE 16. Three-level separable wavelet decomposition of the test image shown in Figure 9. For each level the upper left subimage corresponds to the low-pass coefficients, the upper right subimage corresponds to the vertical coefficients, the lower left subimage corresponds to the horizontal coefficients, and the lower right subimage corresponds to the high-pass residual coefficients.

let decomposition, we refer to the latter as the conventional wavelet decomposition. The decomposition in Figure 17(c) and the Gabor decomposition are similar because frequency space is sampled linearly. However, the wavelet-packet-transform basis functions shown in Figure 18, corresponding to the decomposition in Figure 17(c), are quite different from the Gabor basis functions shown in Figure 6.

Several methods have been proposed for choosing the best representation for a given signal. One of the first methods was to use the energy, log energy, norm, or "entropy" of the coefficients [17, 18]. The decision to continue on to the next level of decomposition is made by comparing the energy of the coefficients in the current level, or node, to that of the coefficients in all descendent components in the next level. In this manner, we can achieve a minimum-energy decomposition. A similar technique was used with an entropy-like criterion to achieve a kind of minimum-entropy decomposition. K. Ramchandran and M. Vetterli improved the selection criterion by minimiz-

ing both rate and distortion [19]. Figure 19 shows a Gabor-like wavelet packet decomposition of the test image shown in Figure 9. It turns out that the minimum-energy and minimum-entropy wavelet packet decompositions of this same test image are identical to the wavelet decomposition.

We have applied these techniques to SAR imagery and found that different images and sensors require different decompositions. Unfortunately, wavelet packet decompositions based on minimizing energy or minimizing distortion for a given rate do not necessarily provide the best reconstructed images from a perceptual point of view. In our image-quality evaluations, we found that the Gabor-like wavelet packet decomposition tends to provide better reconstructed images for SAR imagery and complex visible imagery than the conventional wavelet decomposition. For
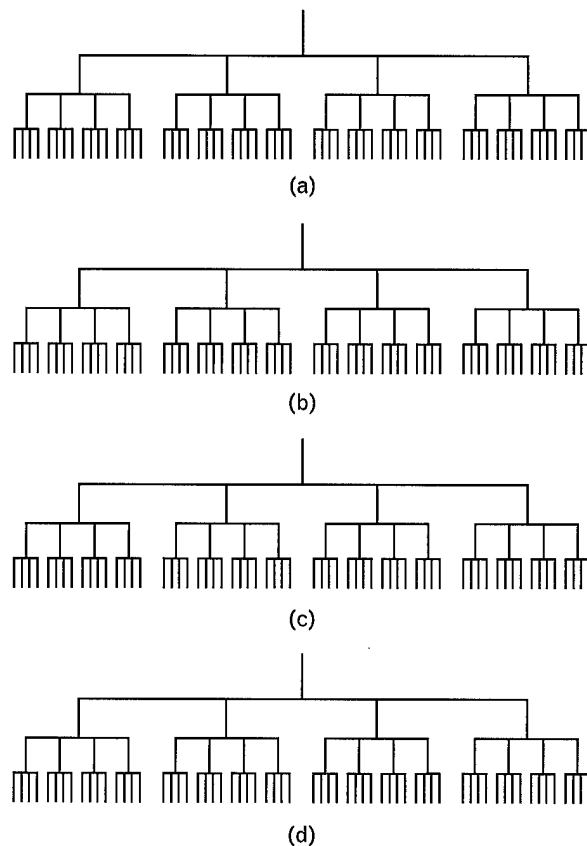
FIGURE 17. Wavelet packet decompositions. Many different bases can be chosen from (a) the full wavelet packet tree. Examples (shown in red) include (b) a subtree corresponding to a conventional wavelet basis, (c) a subtree corresponding to a Gabor-like basis, and (d) an arbitrary subtree.
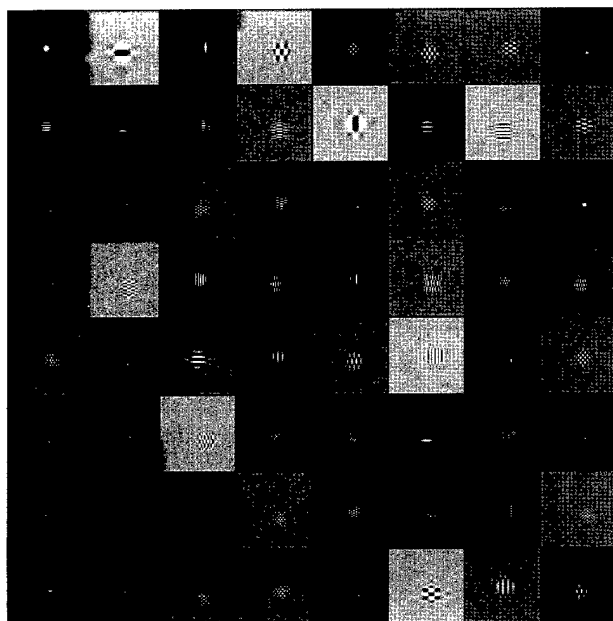
FIGURE 18. Wavelet-packet-transform basis functions corresponding to a Gabor-like decomposition, as shown in part *c* of Figure 17.

FIGURE 19. Three-level wavelet packet decomposition of the test image shown in Figure 9, using a Gabor-like subtree.

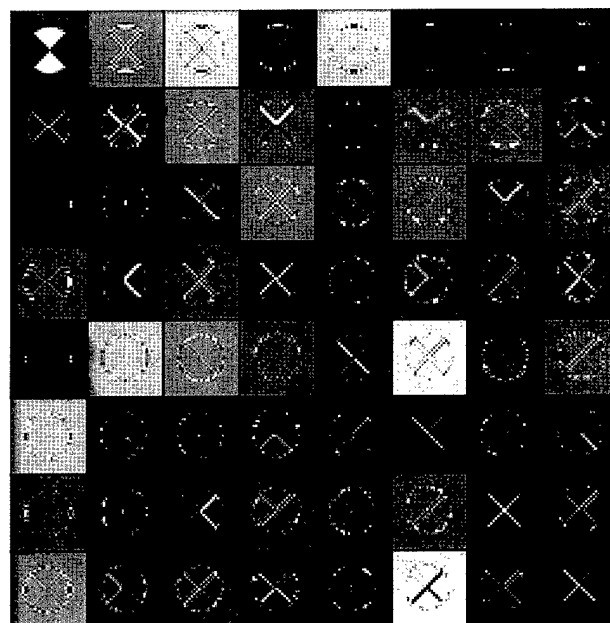many images derived from visible or infrared sensors, decompositions closer to the conventional wavelet decomposition tend to provide reconstructed images with the highest perceptual quality. Here we focus mainly on the Gabor-like wavelet packet decomposition and the conventional wavelet decomposition.

Summary of Transforms

We have compared three types of spatial-spectral transforms: the HVS-like Gabor wavelet transform, the Gabor transform, and the separable wavelet packet transform. The separable wavelet packet transform is a generalization of the separable wavelet transform that includes many possible bases from which to choose, including both a separable wavelet basis, as shown in Figure 17(b), and a Gabor-like basis (the WP-Gabor transform), as shown in Figure 17(c). The major distinguishing attributes of the transforms are the spatial scale selectivity, the orientational selectivity, the number of coefficients used to represent the image, and the computational requirements. The Gabor wavelet transform requires $n_\theta N$ coefficients to represent the image, whereas the other transforms use exactly N coefficients, where N is the number of pixels and n_θ is the number of orientations.

The computational requirements (in terms of number of operations per second) of the Gabor wavelet transform is high relative to that of the other transforms because an iterative procedure is required to determine the coefficients (for example, see Reference 10). Although the complexity of the separable wavelet transform is of order N (where N is the number of pixels in the image), and the complexity of the Gabor transform and complexity of the separable wavelet packet transform with a Gabor-like tree (the WP-Gabor transform) are of order $N \log N$, the actual number of operations required to compute the WP-Gabor transform of a 1024 × 1024 image is only slightly higher than that required for the separable wavelet transform. Table 1 summarizes the attributes and computational requirements of these transforms.

The ability of the Gabor, WP-Gabor, and wavelet transforms to reconstruct the original image accurately at low bit rates can be simulated by setting Gabor coefficients with absolute values that fall below a threshold to zero (because these coefficients will be quantized to zero). For the Gabor, WP-Gabor, and wavelet transforms, we chose thresholds such that 90% of the coefficients were set to zero. For lower threshold values, the differences between the recon-

FIGURE 20. Comparison of an original image and three reconstructed images after thresholding 90% of the Gabor coefficients and remapping the histogram of the three reconstructed images to that of the original image. (a) The original image (of a vehicle on a road in a rural area) was derived from the Lincoln Laboratory Advanced Detection Technology sensor. The other three images were reconstructed from (b) the thresholded Gabor transform, (c) the thresholded separable wavelet transform, and (d) the thresholded separable wavelet packet transform using a Gabor-like tree (the WP-Gabor transform). Most subjects indicate the image in part *d* has the least disturbing artifacts and is most similar to the original image.

structed images are much harder to distinguish. Since the tonal balance, contrast, and brightness of the reconstructed images change after thresholding a large percentage of coefficients, the histograms of these reconstructed images were remapped so that their his-

tograms approximately matched that of the original.

We applied this thresholding and remapping procedure to several SAR images. Figure 20 shows the original image and three reconstructed images for a SAR image (of a vehicle on a road in a rural area) de-

**Table 1. Comparison of the Gabor Wavelet, Gabor, Separable Wavelet,
and WP-Gabor Transforms**

Transform	Orientation Selectivity	Multiple Spatial Scales	Number of Coefficients	Computational Requirements
Gabor wavelet	Good	Yes	$n_\theta N$	High
Gabor	Good	No	N	Low+
Separable wavelet	Poor	Yes	N	Low
WP-Gabor	Poor	Yes	N	Low+

rived from Lincoln Laboratory's Advanced Detection Technology sensor [20]. The image reconstructed from the thresholded wavelet transform had the lowest mean-square error, the image reconstructed from the thresholded WP-Gabor transform had the next lowest mean-square error, and the image reconstructed from the thresholded Gabor transform had the highest mean-square error. In this case, the image quality (as judged by several subjects with experience in viewing SAR imagery) is closely related to the mean-square error. Table 2 summarizes the results of applying this thresholding and remapping process to several SAR images. In general, the image quality of the reconstructed images based on the thresholded wavelet and WP-Gabor transforms was significantly higher than that of the Gabor transform when 90% of the coefficients were set to zero.

Quantization

The task of the quantizer is to convert high-precision (high bit rate) transform coefficients into a smaller set of symbols with lower entropy. This section compares scalar, vector, and trellis-coded quantizers. Frequency variant and spatially variant grouping strategies for quantizing transform coefficients are also discussed.

Scalar Quantizers

Scalar quantizers are the simplest quantizers because they quantize only a single coefficient at a time. Consequently, their computational complexity is quite low and there is no significant processing delay. Uniform scalar quantizers are the simplest scalar quantizers because the decision and reconstruction levels are equally spaced. Uniform scalar quantizers are optimal with respect to a squared-error distortion criterion if the data are uniformly distributed. Unfortunately, coefficients tend to come in groups with histograms that depart significantly from a uniform distribution. When the coefficient histogram is nonuniform, the quantizer reconstruction levels need to be adjusted, or adapted, to minimize the distortion. Finding the minimum distortion reconstruction levels for a scalar quantizer is very similar to optimal clustering in one dimension.

**Table 2. Comparison of the Image Quality of Reconstructed Images after Thresholding
90% of the Transform Coefficients for Several SAR Images**

Transform	Mean-Square Error	Isolated-Point-Return Quality	Shadow Quality	Texture Quality
Gabor	High	Poor	Poor	Poor
Separable wavelet	Low	Good	Good	Fair
WP-Gabor	Low	Good	Good	Good

The most popular adaptive scalar quantizer (ASQ) is the Lloyd-Max quantizer [21, 22]. The Lloyd-Max algorithm for quantizer design uses an iterative technique to adjust the quantizer reconstruction levels so the distortion is minimized. Entropy-constrained scalar quantization minimizes the distortion for a given rate constraint [23].

To avoid the computational burden of optimizing scalar quantization levels for each different set of coefficients, a statistical modeling approach can be used to precompute optimal quantizer levels for a set of predefined coefficient probability density functions (PDF). A generalized Gaussian PDF has been used by several authors to characterize coefficient histograms [24]. The generalized Gaussian PDF has the form

$$p(x) = \frac{\rho}{2\sigma\Gamma(1/\rho)} \sqrt{\frac{\Gamma(3/\rho)}{\Gamma(1/\rho)}}$$
$$\times \exp\left\{ -\left[\sqrt{\frac{\Gamma(3/\rho)}{\Gamma(1/\rho)}} \left(\frac{|x|}{\sigma} \right) \right]^{\rho} \right\},$$

where $\Gamma(\cdot)$ is the Gamma function. The parameter ρ controls the shape of the distribution. For small values of ρ the density is peaked, whereas for large values of ρ the density has a flattened peak about the mean value. The Laplacian PDF corresponds to $\rho = 1$ and the Gaussian PDF corresponds to $\rho = 2$.

Mallat [15] devised a method of estimating ρ for a data sequence by solving

$$\hat{\rho} = F^{-1}\left(\frac{m_1}{\sqrt{m_2}} \right),$$

where

$$F(\rho) = \frac{\Gamma(2/\rho)}{\sqrt{\Gamma(1/\rho)\,\Gamma(3/\rho)}},$$

and where m_1 is the mean absolute value and m_2 is the second moment. Using Mallat's technique, we have found that coefficient histograms for wavelet packet transforms and Gabor transforms for a variety of different types of imagery can be modeled by generalized Gaussian PDFs with ρ ranging from 0.5 to 4. To ap-

ply this statistical modeling approach to the scalar quantization problem, we chose the best-fitting generalized Gaussian PDF from a set of eight models corresponding to

$$\rho = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}.$$

Optimal scalar quantizers are predesigned for these eight different models by using entropy-constrained scalar quantization.

Vector Quantizers

A vector quantizer (VQ) assigns labels to blocks of coefficients. According to C.E. Shannon's rate-distortion theory [25], quantizing and encoding blocks of coefficients always yields a lower average distortion for a given bit rate. In fact, A. Gersho and R.M. Gray claim that vector quantization is "the ultimate solution to the quantization of a signal vector" and "no other coding technique exists that can do better" [26] (p. 313). The basis of this statement lies in the fact that if we are allowed to select an appropriate codebook, vector quantization can perform as well as any other technique. What is not said is that the memory capacity required to store the codebook may be impractical, and the time required to search the codebook to find the most appropriate codevector may also be impractical. Thus the key issues in designing a VQ-based coder are (1) block size, (2) codebook design, and (3) the codebook search scheme.

VQs with the highest complexity are referred to as full-search, flat, or unconstrained VQs. The most popular VQ algorithm is a type of clustering technique known as the LBG algorithm [27], so named after the initials of the authors' last names. If R bits per vector are used with n-dimensional vectors, the complexity of a flat VQ is of order 2^{nR} because there are 2^{nR} possible codevectors that must be searched for the best match to the input vector.

There are three major ways to reduce the complexity of a VQ for a given rate: (1) reduce the block size (decrease n), (2) add structure to the codebook, or (3) reduce the size of the codebook. Unfortunately, each of these solutions decreases the accuracy or generality of the quantizer. Tree-structured VQs increase the efficiency of the search by adding structure to the

codebook. Limited-size VQ codebooks are usually designed by collecting a set of typical images of interest, called the training set. The training set is used to design a set of codevectors that minimize a distortion criterion. Entropy-constrained VQs minimize a two-sided distortion criterion that includes both distortion and rate [23]. (See the section entitled "Bit Allocation" for a more thorough discussion of minimizing distortion and rate simultaneously.) There are many varieties of VQs; for a thorough discussion of the various types of VQs applied to compression, see Reference 26.

Trellis-Coded Quantizers

Trellis-coded quantization is an efficient type of tree-structured vector-quantization method with low to moderate complexity, compared to full-search VQs. In fact, trellis-coded quantization represents a compromise between the efficiency of scalar quantization and the accuracy of full-search vector quantization.

A trellis was originally conceived as a state-transition diagram for a finite-state machine. It can be thought of as a tree with a restricted branching structure. The name *trellis* is based on the resemblance of a restricted tree-evolution diagram (which depicts the paths taken through the tree structure during the decoding operation) to a common garden trellis [26]. Trellis coding is a proven, asymptotically optimal (in a rate-distortion sense) technique for source coding [28, 29], but most implementations use a stochastically populated trellis and have high computational complexity. Trellis-coded quantization is an efficient approach that uses a structured and deterministic scalar codebook and a restricted trellis structure.

Consider the quantization of a set of real numbers (possibly transform coefficients) into a restricted set of output (reconstruction) levels. Each real number is to be quantized into one output level. In scalar quantization, each input number is quantized independently from other input numbers. In vector quantization, however, the set of output symbols emitted by the quantizer depends on all input numbers within a block. Trellis-coded quantization considers the entire sequence and finds the best (e.g., minimum squared error) path through the trellis, which is equivalent to finding the sequence of allowable output levels that minimizes the distortion. The trellis path lengths are determined by the distance between the current input number and each output level. The Viterbi algorithm [30], described in the sidebar entitled "The Viterbi Algorithm," is used to find the minimum distortion path through the trellis.

To illustrate the concept of trellis-coded-quantization, we consider the input sequence $\{-2.5, 0.5, 3.5\}$ and the four-state trellis and subset labeling shown in Figure 21. We assume that the initial state is S_2 (the third node from the top in the figure). Note that only two paths emanate from each state, and each subset (D_0, D_1, D_2, and D_3) contains only two codewords. Thus only two bits are required to specify each element in the output sequence; i.e., the output bit rate is two bits per sample. The trellis path lengths are assigned according to the distance (absolute difference) between the input numbers and the codewords. The minimum distance (minimum squared error) output sequence is $\{-3, 1, 5\}$, which has a squared error of 2.75. The trellis-coded-quantization output bit sequence corresponding to this output sequence is $\{00, 01, 11\}$.

Both fixed-rate and variable-rate trellis-coded-quantization systems can be designed. M.W. Marcellin [31] and Marcellin and T.R. Fischer [32] discussed fixed-rate trellis-coded quantization, but more recently Fischer and M. Wang [33] and Marcellin [34] introduced a variable-rate trellis-coded quantization called entropy-constrained trellis-coded quantization in order to achieve better performance at lower bit rates. These entropy-constrained trellis-coded-quantization approaches require a computationally intensive procedure for designing the codebook.

Two simpler alternatives have been developed that do not require a computationally intensive codebook design procedure. R.L. Joshi, V.J. Crump, and Fischer developed a trellis-coded-quantization procedure based on uniformly spaced codewords [24]. J.H. Kasner, Marcellin, and B.R. Hunt developed a more accurate alternative with a negligible increase in complexity, which they call universal trellis-coded quantization (UTCQ) [35]. UTCQ does not require large codebooks or a computationally intensive codebook design procedure. Instead, the encoder computes certain coefficient statistics that are used by the decoder

THE VITERBI ALGORITHM

THE VITERBI ALGORITHM is an optimal recursive solution to the problem of finding the shortest path through a trellis. In the context of trellis-coded quantization, the Viterbi algorithm is used to find the path through the trellis that minimizes the squared error between a given input sequence and all possible quantized output sequences.

To describe the use of the Viterbi algorithm within the context of a trellis-coded quantizer, we consider the trellis and subset labels of Figure 21. We start in state S_2. The first number in the input sequence is -2.5. The upper segment emanating from S_2 corresponds to choosing a codeword from subset D_2. The two codewords in subset D_2 are -3 and 5; thus the closest codeword in subset D_2 to the input number is -3. We assign segment lengths in the trellis according to the absolute difference between the closest codeword and the input number, so that the upper

segment emanating from S_2 is assigned a length of 0.5. Similarly, the lower segment is assigned a length of 3.5 because the closest codeword in subset D_0 is 1. This process is continued for all segments until the end of the input sequence, and of each path, is reached.

With such a short sequence, the shortest path can be easily and quickly found by computing the lengths of each of the eight possible paths. If the sequence is significantly longer, however (or if the trellis is less restricted), computing the lengths of all possible sequences can quickly become prohibitively expensive.

The first person to address the problem of finding the shortest path through a graph of this type was G.J. Minty [1, 2]. Minty's solution was to build a string model of the trellis with knots at each state node and string lengths corresponding to the segment distances. Then with the initial state anchored, the knots

at the final states are pulled tight. The shortest path is indicated by the strings that stretch most tightly when pulled. Unfortunately, this solution is not appropriate for long sequences, nor is it easy to program on a computer.

In studying convolutional codes and their corresponding decoding algorithms, Viterbi identified a forward dynamic programming solution [1]. Viterbi's key insight was to recognize that at any time k the shortest paths to each node can be computed, and the shortest path through the trellis must pass through one of these nodes. Thus the shortest path can be computed recursively by extending each shortest path from the current node at time k to time $k + 1$ and computing the shortest path to each node at time $k + 1$. (Time here really means sequence item. The time index k corresponds to the kth item in the sequence. We speak of time k because we are stepping through the trellis se-

to achieve nearly optimal reconstruction levels for the first two codewords on each side of zero. These four codewords add negligible side information to the compressed bit stream. The rate distortion (mean-square error) performance of UTCQ on natural imagery is competitive with the best techniques known [16, 36–39], and we think it represents the best trade-off between accuracy and complexity among current techniques.

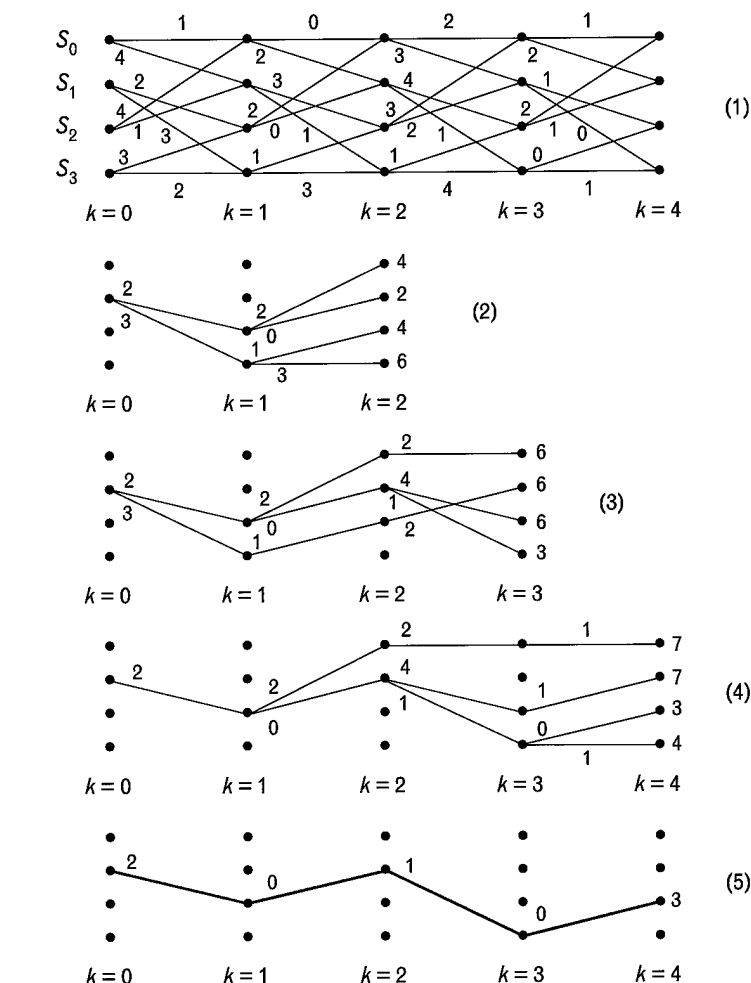*Quantizing Transform Coefficients*

Given a spatial-spectral transform, such as the Gabor transform or the WP-Gabor transform, the following design question arises. How should the coefficients be grouped for quantization? The simplest approach is to quantize all coefficients with the same quantizer. However, if the coefficients have different statistics or varying significance, then grouping the coefficients

quentially in time.) The number of shortest paths that must be stored never exceeds the number of states.

Figure A illustrates the recursive determination of the shortest path via the Viterbi algorithm. This figure is an example of a four-state trellis with segment lengths as labeled. We assume we start at state S_1. For the first two time units we extend the paths to each of the four nodes and compute the lengths of the paths to each node, as shown in Figure A(2). We extend each of these paths to the next time unit by selecting the shortest path to each node at time $k = 3$, given the previously computed path lengths up to time $k = 2$ and the segment lengths from time $k = 2$ to time $k = 3$. The shortest paths to times $k = 3$ and $k = 4$ are shown in Figures A(3) and A(4), and Figure A(5) shows the shortest path through the trellis.


FIGURE A. Illustration of the recursive determination of the shortest path through a trellis by using the Viterbi algorithm. (1) Four-state trellis with segment lengths as labeled. The initial state is S_1 (the second node from the top). (2) Paths from time $k = 0$ to $k = 2$. The path lengths are given at the end of each path. (3) Shortest paths from time $k = 0$ to each node at time $k = 3$, and their path lengths. (4) Shortest paths from time $k = 0$ to each node at time $k = 4$ and their path lengths. (5) The shortest path through the trellis.

References

1. G.D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE* **61**, Mar. 1973, pp. 268–278.
2. G.J. Minty, "A Comment on the Shortest-Route Problem," *Oper. Res.* **5**, Oct. 1957, p. 724.

according to their statistics or their significance and using more than one quantizer tends to yield better performance (lower distortion for the same rate). Since coefficients associated with different frequency bands tend to have different statistics and different perceptual significance, one such approach is to assign each different frequency component to a different quantizer. This approach groups coefficients belonging to the same frequency band and is called frequency-variant quantization, or subband quantization. Another approach is to group coefficients belonging to the same spatial region and assign different spatial regions to different quantizers. This approach is called spatially variant quantization. Approaches that combine both strategies are referred to as spatial-frequency quantization (SFQ).

J.M. Shapiro developed one of the first SFQ methods using the separable wavelet transform [36]. Spa-

(a)

(b)

(c)

(d)
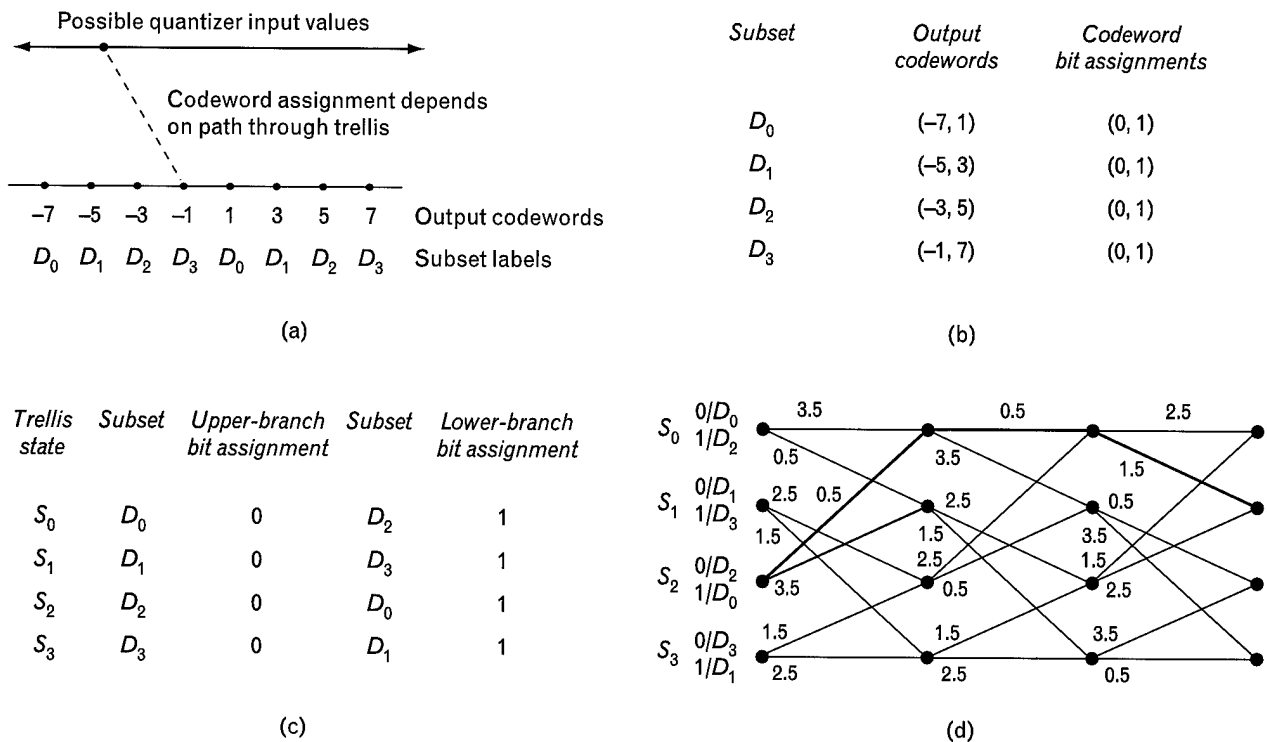
FIGURE 21. A four-state trellis with subset labeling and the correspondence to the quantizer outputs. (a) The codewords indicate the possible output values from the quantizer. The quantizer assigns output codewords and associated subset labels based on the path through the trellis. Numbers input to the quantizer can lie anywhere on the line. (b) Output codewords, their subset labels, and their bit assignments. (c) Trellis states and their corresponding subset labels and bit assignments. (d) Four-state trellis with emanating path labels. The path labels indicate the trellis path taken and the subset from which the codeword was chosen. For example, from trellis state S_2 (third node from the top), $0/D_2$ indicates the upper path was taken (with bit assignment 0) and the codeword was chosen from subset D_2. If the input value is -2.5, then the closest codeword is -3, so the emitted bit sequence is 00. The minimum squared-error path corresponding to the input sequence {-2.5, 0.5, 3.5} is indicated in red (assuming initial state S_2). The trellis-coded-quantization output sequence is {-3, 1, 5} and the emitted bit sequence is {00, 01, 11}.

tial quantization was achieved via wavelet zerotrees, which are based on the hypothesis that if a coefficient at a coarse spatial scale is quantized to zero, then all coefficients of the same orientation in the same spatial location at finer scales should also be quantized to zero. Zerotrees efficiently encode values of multiscale coefficients by using a single symbol to represent the set of all coefficients with the same orientation in the same spatial location that are quantized to zero with this hypothesis.

Z. Xiong, Ramchandran, and M.T. Orchard developed an improved but more computationally intensive SFQ procedure for wavelet transforms. This SFQ procedure optimizes, in a rate-distortion sense, the trade-off between zerotree encoding and scalar quantizing coefficients [39]. Their SFQ approach has been used for compressing natural imagery, and their performance is among the best reported in the literature [16, 36–38]. The performance of their technique on SAR images has yet to be determined. There is some indication that SFQ tends to despeckle SAR images, but, as mentioned previously, it is not clear that despeckling is desirable because despeckling degrades the background texture.

Summary of Quantizers

We have discussed scalar quantizers, vector quantizers, trellis-coded quantizers, and spatial-frequency quantizers. Table 3 compares these quantizers with respect to computational requirements, training requirements, storage requirements, and the ability to reduce spatial redundancy.

Bit Allocation

Given a set of N quantizers, the bit-allocation problem is to assign bit rates r_i to each quantizer such that the overall coefficient distortion D is minimized, subject to the constraint that the total bit rate is less than or equal to some specified target bit rate R; in other words, the problem is to minimize

$$D = \sum_i^N D_i(r_i)$$

subject to

$$\sum_i^N r_i \leq R,$$

where $D_i(r_i)$ is the distortion incurred by the ith quantizer at rate r_i. Note that this formulation minimizes the distortion in the coefficient domain, as opposed to the image domain. Orthogonal transforms are variance preserving [40], so minimizing distortion in the coefficient domain is equivalent to minimizing distortion in the image domain for orthogonal transforms. However, for nonorthogonal transforms such as the Gabor transform, minimum distortion in the coefficient domain does not guarantee minimum distortion in the image domain.

Minimizing Distortion with a Bit-Rate Constraint

Dynamic programming approaches can be used to obtain solutions to the bit-allocation problem. However, the complexity of these approaches is high—on the order of B^2 operations per quantizer if the total number of bits is B. For some quantizers at high rates, $D_i(r_i)$ can be approximated as

$$D_i(r_i) \approx \varepsilon_i^2 \sigma_i^2 2^{-2r_i},$$

where σ_i^2 is the variance of the coefficients quantized by the ith quantizer and ε_i is a performance factor that depends on the probability density of the coefficients and on the type of encoding. The bit-allocation solution derived from this approximation, using the method of Lagrange multipliers [40], is

$$R_i = R + \frac{1}{2} \log_2 \frac{\varepsilon_i^2 \sigma_i^2}{\left(\Pi_j^N \varepsilon_j^2 \sigma_j^2\right)^{1/N}}.$$

Table 3. Summary of Quantizer Characteristics

Quantizer*	Storage Requirements	Training Required	Computational Requirements**	Spatial-Redundancy Reduction
Uniform scalar quantizer	None	No	None/low	None
Adaptive scalar quantizer	Scalar codebook	Yes	Moderate/low	None
Entropy-constrained scalar quantizer	Scalar codebook	Yes	High/low	None
Universal trellis-coded quantizer	Negligible	No	None/moderate	None
Entropy-constrained trellis-coded quantizer	Scalar codebook	Yes	High/moderate	None
Vector quantizer	Vector codebook	Yes	High/moderate to high	Yes
Spatial-frequency quantizer using a uniform scalar quantizer (SFQ/USQ)	None	No	None/high	Yes

*   With the exception of SFQ/USQ, the quantizers are listed in order of increasing quantization accuracy for a given bit rate when tested on pseudo-randomly generated sequences with generalized Gaussian probability density functions ($\rho = 1, 2$). SFQ/USQ is designed to work with spatially meaningful data such as a set of coefficient subimages.
** Computational requirements are listed for training and operational modes (training/operational).

As we might expect, the high-rate approximation for $D_i(r_i)$ tends to break down at low rates for most quantizers. Y. Shoham and A. Gersho devised a bit-allocation procedure that works for arbitrary $D_i(r_i)$ [41]. Their algorithm efficiently determines the rate-allocation vector

$$B^* = (r_1^*, r_2^*, \ldots, r_n^*),$$

which is a solution to

$$\min_B \left\{ \sum_{i=1}^{N} D_i(r_i) + \lambda \sum_{i=1}^{N} r_i \right\}, \quad (4)$$

subject to

$$\sum_{i=1}^{N} r_i \leq R \text{ and } a_i \leq r_i \leq b_i, \quad (5)$$

where the admissible rates are bounded by a_i and b_i and λ is a Lagrangian multiplier that can be thought of as the rate cost factor. With $\lambda = 0$ the cost function is independent of the bit rate; as $\lambda \rightarrow \infty$ the cost function is essentially independent of the quantizer distortions $D_i(r_i)$. The solution to the constrained problem (Equations 4 and 5) is also a solution to the unconstrained problem (Equation 4 only). Shoham and Gersho's major insight was that the unconstrained Equation 4 can be minimized by minimizing each term in the sums separately. That is, the solutions r_i^* to

$$\min \left\{ D_i(r_i) + \lambda r_i \right\}$$

for each $i = 1, 2, \ldots, N$ comprise a solution to Equation 4.

As previously mentioned, the bit-allocation formulation does not guarantee that the reconstructed image distortion is minimized for nonorthogonal transforms. However, a large reduction in the distortion in the coefficient domain tends to reduce the distortion in the image domain.

Perceptually Weighted Distortion

If a mean-square-error distortion metric is used, as is typically the case in the image-compression literature, the *perceived* distortion is not necessarily minimized. For frequency-variant quantization methods, the per-ceived distortion is related to the spatial-frequency characteristics of the human visual system (HVS). The HVS is more sensitive to low spatial frequencies than to high spatial frequencies, as shown in Figure 22(a). This reduced sensitivity to high spatial frequencies produces a type of psychovisual redundancy.

Figure 22(a) shows the HVS spatial-frequency sensitivity function, which is sometimes referred to as the contrast sensitivity function, because the data in Figure 22(a) are obtained by measuring thresholds at which subjects can just noticeably detect changes in the contrast of a sinusoidal grating stimulus as a function of spatial frequency. Although the contrast sensitivity function tells us something about the spatial-frequency sensitivity of the HVS, we must keep in mind that it depends not only on controllable factors such as ambient lighting and viewing distance, but also on local context. For example, D.H. Kelly showed that the presence of edges can dramatically change the contrast sensitivity function [42]. Nevertheless, we can incorporate the HVS spatial-frequency sensitivity characteristics into the bit-allocation procedure by simply weighting the coefficient distortion corresponding to each frequency band according to the contrast sensitivity function. In this case, the total distortion becomes

$$D = \sum_i^N w_i D_i(r_i),$$

where the coefficients w_i are referred to as perceptual weighting factors.

Psychophysical experiments [43] suggest that the HVS spatial-frequency sensitivity function has the form

$$A(f_r) = \left[a + (f_r/b)^{c_1} \right] e^{-(f_r/b)^{c_2}},$$

where f_r is the radial frequency in cycles per degree of visual angle. The form of $A(f_r)$ was derived from psychophysical experiments in which the contrast sensitivity of sinusoidal gratings as a function of frequency was measured. Although the form of $A(f_r)$ can change for different contexts, it has been used successfully for bit allocation by several authors for optical imagery [44, 45]. We use the parameter values $a =$
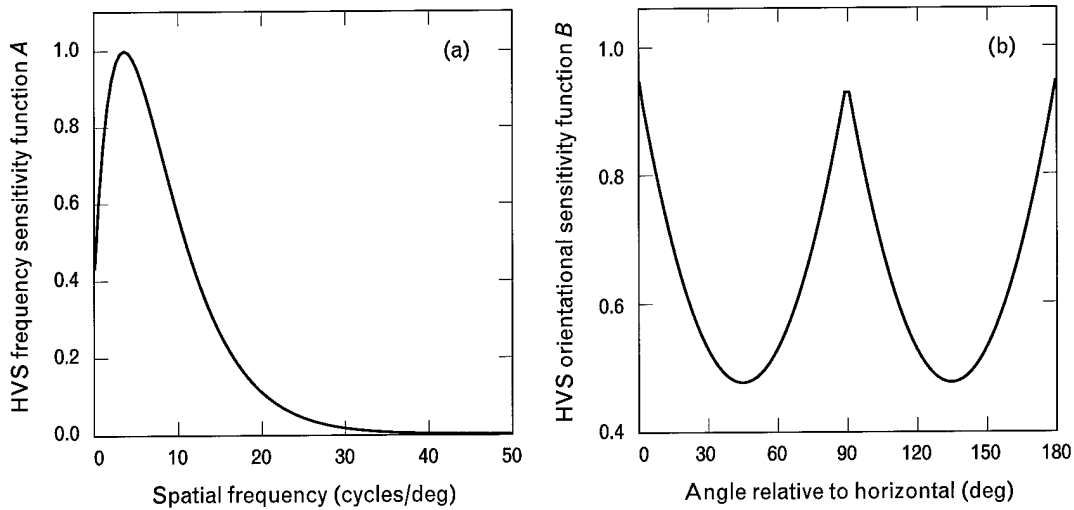
FIGURE 22. (a) Estimated spatial-frequency sensitivity function of the human visual system (HVS), indicating less sensitivity to high spatial frequencies. (b) Estimated orientational sensitivity function of the HVS, indicating greater sensitivity to distortions in the horizontal and vertical directions.

0.192, $b = 4.38$, and $c_1 = c_2 = 1$, which is quite similar to the function used by R.H. Bamberger and M.J.T. Smith [45], and compare this bit-allocation scheme to the distortion-rate scheme described previously.

It is important to point out that $A(f_r)$ depends on the visual angle subtended by the image, i.e., on the image size and the viewing distance. For example, the harmonic frequencies used in the Gabor expansion can be converted to cycles per degree of visual angle by determining the number of cycles per degree for the highest frequency.

Psychophysical experiments have shown that the HVS is more sensitive to distortions oriented in the horizontal and vertical directions than other directions [46]. Physiological experiments indicate, however, that this orientation sensitivity changes dramatically for different contexts [47]. The section entitled "Evaluating Compression System Performance" describes an experiment in which images were compressed with the Gabor transform, using perceptual weights with no orientational variation (isotropic) and with the following orientational sensitivity function, consistent with M.M. Taylor's data [45, 46]:

$$B(f_\theta) = \begin{cases} \dfrac{8}{\pi^2} f_\theta^2 - \dfrac{4}{\pi} |f_\theta| + 1, & \text{for } 0 \le |f_\theta| \le \dfrac{\pi}{2} \\ \dfrac{8}{\pi^2} f_\theta^2 - \dfrac{12}{\pi} |f_\theta| + 5, & \text{for } \dfrac{\pi}{2} < |f_\theta| \le \pi \end{cases}$$

Figure 22(b) shows the function $B(f_\theta)$; f_θ denotes the (spatial frequency) angle relative to the horizontal.

In the experiment described in the section entitled "Evaluating Compression System Performance," the number of bits per pixel (bpp) allocated to each frequency component for the contrast-sensitivity-function-based bit-allocation method was determined by using

$$R_{rs} = \frac{N_{rs}}{N^2} \log_2(l_{max} w_{rs}),$$

where N is the number of pixels and N_{rs} represents the number of coefficients in the (r, s) frequency component ($N_{rs} = n_p n_q$ for the real components and $N_{rs} = 2n_p n_q$ for the complex components). The w_{rs} are frequency weighting factors corresponding to the product $A(f_r) B(f_\theta)$ evaluated at the (r, s) harmonic. The parameter l_{max} controls the maximum number of quantization levels allocated across all harmonics and is adjusted to yield the desired total bit rate. The total bit rate is determined by summing R_{rs} across all pairs (r, s), except those components which can be determined by conjugate symmetry.

Summary of Bit-Allocation Techniques

The bit-allocation problem can be viewed as a constrained optimization problem in which image distortion and bit rate are jointly minimized. The bit-allo-

cation problem is traditionally viewed as a trade-off between image quality and bit rate, and image quality is considered a nondecreasing function of bit rate. In the next section, however, we show that the minimum distortion for a given bit-rate constraint does not necessarily yield the minimum perceptual distortion. This dilemma is caused by the inconsistency between objective distortion criteria, such as mean-square error, and subjective distortion criteria.

We have discussed two approaches to overcoming this problem. In the rate-distortion minimization framework, the distortion can be perceptually weighted according to the spatial-frequency sensitivity of the HVS. A slightly less computationally intensive approach is to assign quantizer bit rates or, more directly, step sizes according to the HVS spatial-frequency sensitivity. We have tried both approaches and found only minor differences in perceived image quality. We have also found that taking into account a preference for vertical and horizontal spatial-frequency components with this latter method does not improve the perceived image quality.

Evaluating Compression-System Performance

Compression systems can be evaluated by objective or subjective measures. The most appropriate evaluation criteria are dictated by the applications in which the compression system will be used. For visible imagery, we know that the most widely used objective measure for compression-system optimization—mean-square error—is a poor predictor of subjective image-quality ratings. Even so, most researchers use it because it provides a quick and easy measure of performance, and it can be minimized by using various optimization techniques such as the one described in the previous section. In contrast, subjective measures, based on psychophysical experiments involving three to twelve human subjects, are time consuming and expensive. However, if humans are the users of the compressed images, then subjective measures are usually required.

Objective Measures of Performance

As mentioned above, the most commonly used objective performance measure is the mean-square-error metric. Other similar pixel-comparison metrics in-

clude the mean absolute error and the maximum absolute error. Most pixel-comparison metrics fail to correlate with subjective ratings because the HVS appears to weight visual differences according to local context and features of interest. However, there is some evidence that the distortion contrast metric, which represents the average contrast between the intensities of pixels in the original and reconstructed images, tends to correlate with subjective ratings [48].

The normalized mean-square-error (NMSE) metric is defined as

$$\text{NMSE} = \frac{\sum_i (I_i - R_i)^2}{\sum_i I_i^2},$$

and the distortion contrast (DCON) metric is defined as

$$\text{DCON} = \frac{1}{N} \sum_i \frac{|I_i - R_i|}{a + I_i + R_i},$$

where I_i is the intensity value of the ith pixel in the original image, R_i is the intensity value of the ith pixel in the reconstructed image, and N is the number of pixels. The constant a depends on the relationship between luminance and gray level for the display; we used $a = 23/255$.

Several researchers have attempted to use objective measures that do not depend on pixel comparisons. For example, T.D. Penrod and G.G. Kuperman compared a measure based on the –3-dB width of isolated point returns and a region-based contrast ratio measure to subjective ratings, but they found no correlation [49]. N. Chaddha and T.H.Y. Meng developed a psychovisual distortion measure for a limited set of visible images and were able to predict subjective image-quality ordering quite well, but this technique has not been applied to SAR imagery [50].

In an effort directed toward the development of perceptual distortion metrics, we have studied the effect of sharpening isolated point returns and attenuating noise in shadows on the perceived image quality. We found that both of these operations, when applied only to objects or shadows, respectively, tend to increase the perceived image quality. However, we

have yet to translate these operations into reliable metrics for predicting perceived image quality.

Subjective Measures of Performance

The most popular and dependable experimental paradigms for the assessment of image quality include (1) the simultaneous-rating paradigm, (2) the two/three alternative forced-choice (TAFC) paradigm, (3) the flicker paradigm, and (4) task-dependent paradigms. The guiding principle in psychophysical experimentation is to design the experiment to make it easy for the subject to choose, i.e., limit the number of choices and make the choices as simple as possible.

In the simultaneous-rating paradigm, all images are shown to the subject simultaneously and the subject rates the imagery on a five-point or ten-point scale. Only a small number of images can be displayed with this paradigm, and the fewer the better. Because the subject has to compare several images and decide what rating to assign to each image, this paradigm is the least reliable and the most fatiguing of the four. The TAFC paradigm tends to be more reliable than the simultaneous-rating paradigm because only two images have to be compared, and the subject has only two possible choices. A variation gives the subject a third choice (no preference of one image over the other).

In both the simultaneous-rating and TAFC paradigms, the subject must make numerous eye movements within and between images. The accuracy of a subject's choice depends to some extent on the short-term memory associated with retaining the image features that are being compared. The flicker paradigm eliminates the need for making eye movements between images by placing the images in the same spatial location and temporally switching between images (say, at one-second intervals). For this reason, the flicker paradigm tends to be more robust than the first two paradigms. Task-dependent paradigms require the subject to perform a task that is usually associated with the application. For example, a subject might be required to point to an object in a scene, and the accuracy and response time can be used as the subjective performance measures.

Subjective measures are also used to evaluate artifacts in images. At low bit rates, only a few coeffi-
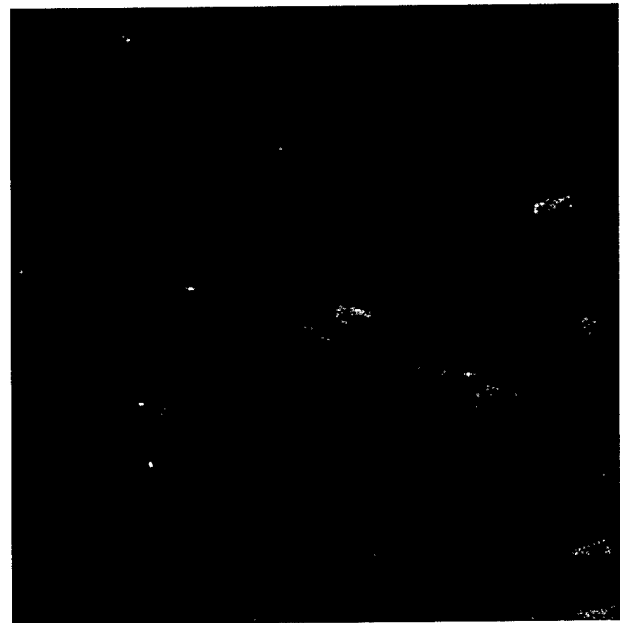

FIGURE 23. Original detected log-amplitude SAR 512 × 512-pixel reference image used as input to the compression system. The spatial resolution in this image is approximately one foot.

cients are used to reconstruct the image in some regions. As a result, transform artifacts that reveal the structure of the basis functions are apparent. Given several transforms, subjective experiments must be performed to determine which transform has the least annoying artifacts. For example, in our evaluations we have found that the Gabor and the Gabor-like wavelet packet decompositions have the least annoying background artifacts in SAR images.

We have conducted several psychophysical experiments, all of which have shown that mean-square error is also a poor predictor of subjective ratings for SAR imagery. First, we present a comparison of three different compression systems, each based on a different representation system. The original image was derived from SAR spotlight-mode imagery collected in Stockbridge, New York, using the Lincoln Laboratory Advance Detection Technology sensor. Figure 23 shows the original detected log-amplitude SAR image (512 × 512 pixels) that was processed by the compression system. The spatial resolution is approximately one foot. Receiver voltages from the in-phase (I) and quadrature-phase (Q) channels were digitized with 20-bit precision (two 8-bit mantissas and one 4-bit
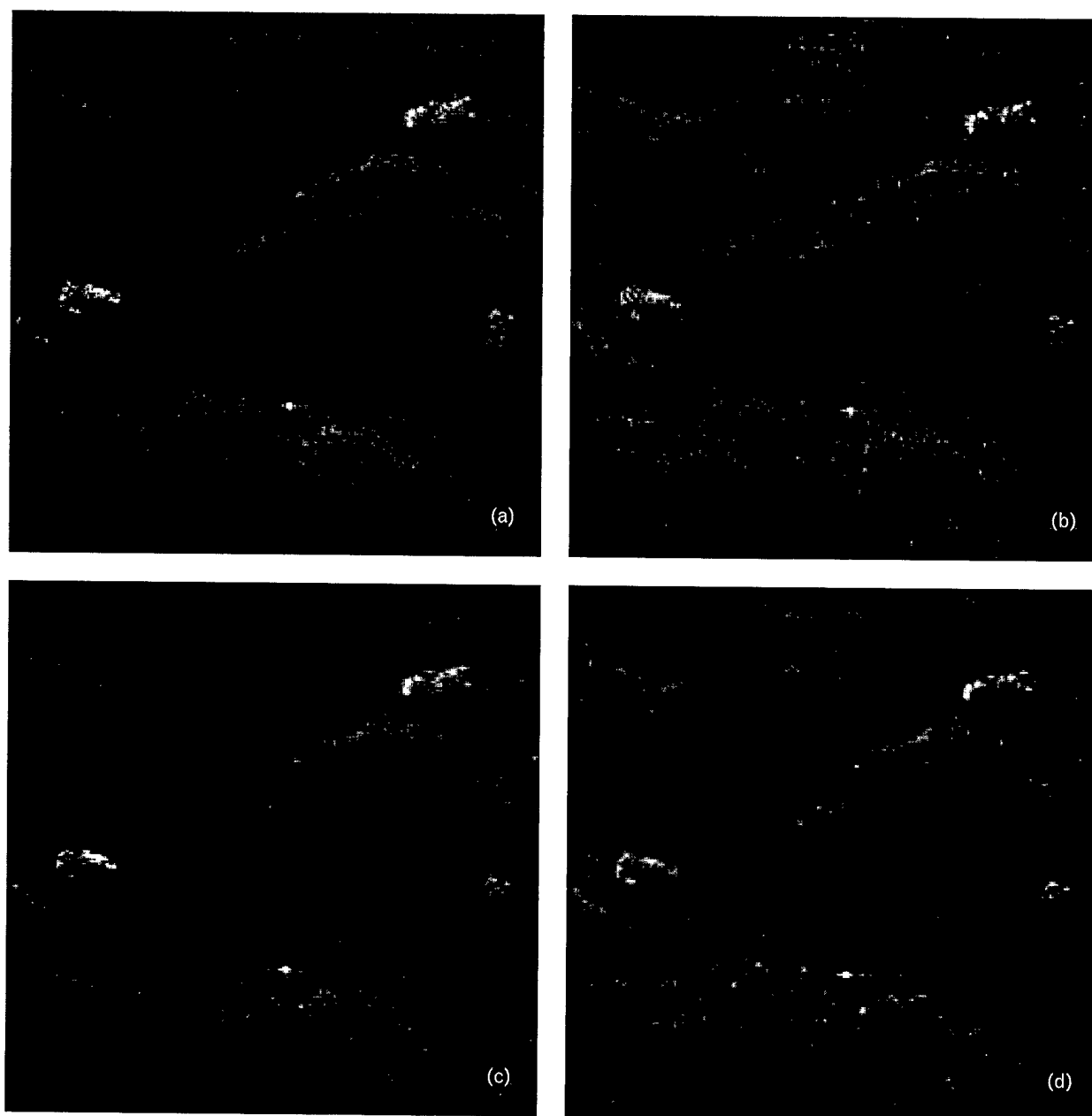
FIGURE 24. (a) Subimage (256 × 256 pixels) of the original SAR image shown in Figure 23. For the original image, the entropy is 14 bits per pixel (bpp), stored as 20 bpp. (b) Reconstructed image using a Gabor-based compression system with a spatial-frequency-weighted distortion metric. (c) Reconstructed image using a separable wavelet-based compression system. (d) Reconstructed image using a separable wavelet packet transform with a Gabor-like basis (WP-Gabor transform). All of the compression systems used trellis-coded quantizers and arithmetic entropy coders. The three reconstructed images were each compressed to 0.5 bpp prior to reconstruction. Most subjects indicate that the image in either part *b* or part *d* is most similar to the original image.

exponent common to I and Q), and the entropy of the log-amplitude image is approximately 14 bpp.

The original image was compressed to 0.5 bpp and then reconstructed. Three different compression systems were used: (1) a Gabor transform system, (2) a separable wavelet system, and (3) a separable wavelet packet transform-based system with a Gabor-like basis (i.e., a WP-Gabor transform-based system). The bit allocator for the Gabor system used a spatial-frequency weighted distortion metric. Without spatial-

**Table 4. Comparison of Objective Measures of Image Quality and
Subjective Ratings for the Images Shown in Figure 24**

| Transform	NMSE*	DCON*	Isolated-Point-Return Quality	Shadow Quality	Texture Quality
Gabor	0.121	0.178	Good–	Good–	Good
Separable wavelet	0.097	0.164	Good	Good	Poor
WP-Gabor	0.101	0.164	Good–	Good	Good

* NMSE is the normalized mean-square-error metric; DCON is the distortion contrast metric.

frequency weighting, the Gabor system was not competitive. Figure 24 shows the original SAR image and three reconstructed images, and Table 4 gives the normalized mean-square-error (NMSE) metric and the distortion contrast (DCON) metric for each image. The objective metrics reveal that the separable-wavelet-based system has the lowest NMSE and DCON values, while the Gabor-based system has the highest values. Table 4 also summarizes the image-quality ratings of four subjects. For most SAR images, the separable-wavelet system tends to preserve the quality of the isolated point returns quite well, but it tends to reconstruct textures poorly. The Gabor and wavelet-packet systems tend to reconstruct textures better than the separable-wavelet system.

Next we present an example of an experiment in which eleven subjects were asked to rate nine SAR images. The purpose of this experiment was to determine which quantization method—adaptive scalar, vector, or trellis-coded—yielded the highest perceived image quality for a Gabor-transform-based compression system. All of the subjects had experience viewing SAR imagery but none were experts in analyzing SAR imagery. The training and test images were derived from SAR spotlight-mode imagery collected in Stockbridge, New York, again using the Lincoln Laboratory Advance Detection Technology sensor. Since the data were collected at different aspect angles, the training and test sets were designed such that the training images consisted of data collected at aspect angles from 0° to 45° and the test image represented data collected at an aspect angle of 60°.

Different quantizers were used to compress the test

images in a Gabor-based compression system with an arithmetic entropy coder. We chose to compress the images to bit rates as close to 0.5 bpp as possible (within 0.05 bpp). Figures 25 through 29 show the test images that correspond to each of the reconstructed images, and Table 5 summarizes the NMSE values and the normalized maximum absolute error (NMXE) values for each of the test images except the original. The JPEG image has the lowest NMSE, and the Gabor/VQ image corresponding to frequency-dependent block sizes has the highest NMSE. The orientation-selective Gabor/TCQ image has the lowest
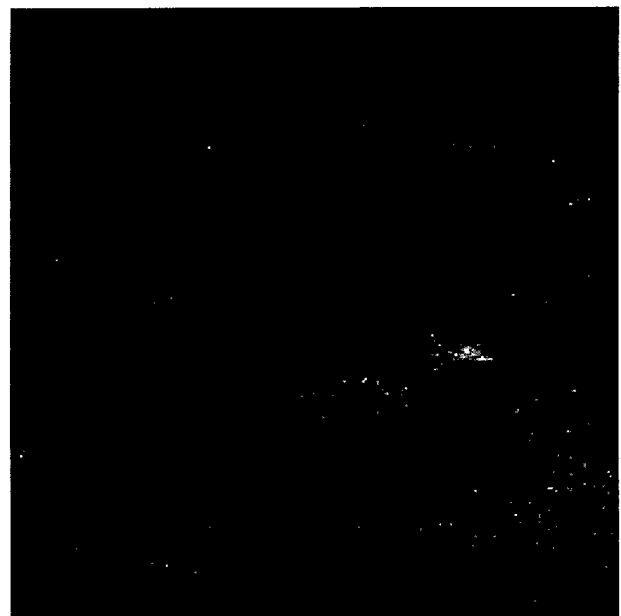
FIGURE 25. Objectionable test image. This image was produced by using an adaptive scalar quantizer with only two levels for each of the modulated (AC) components.
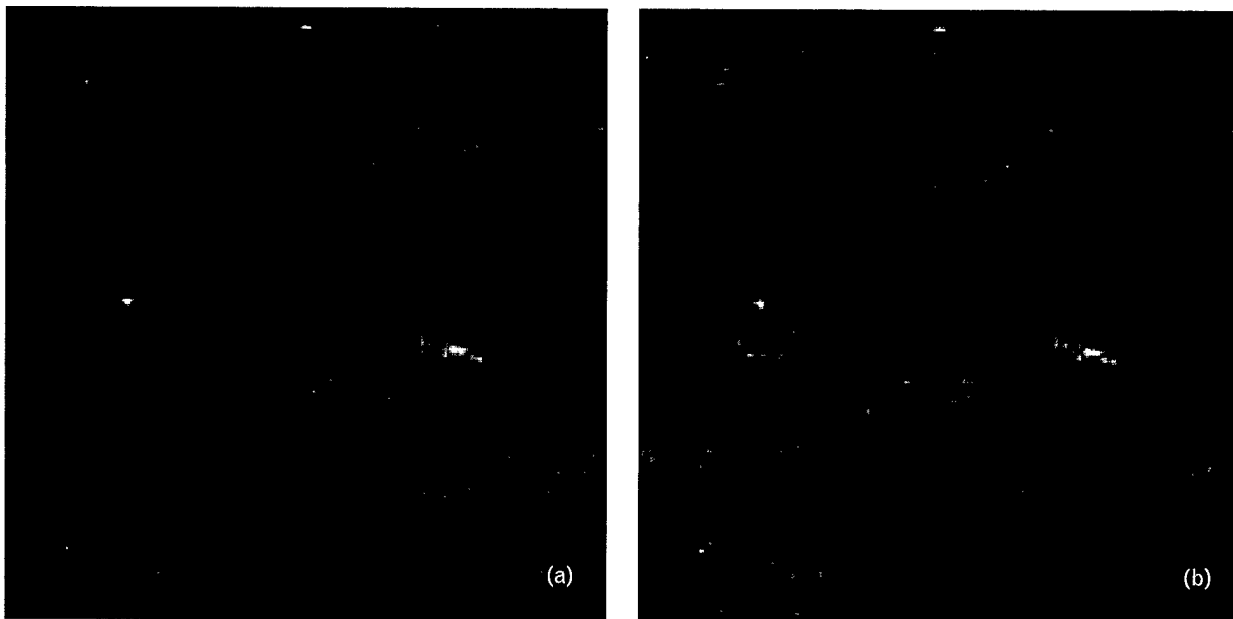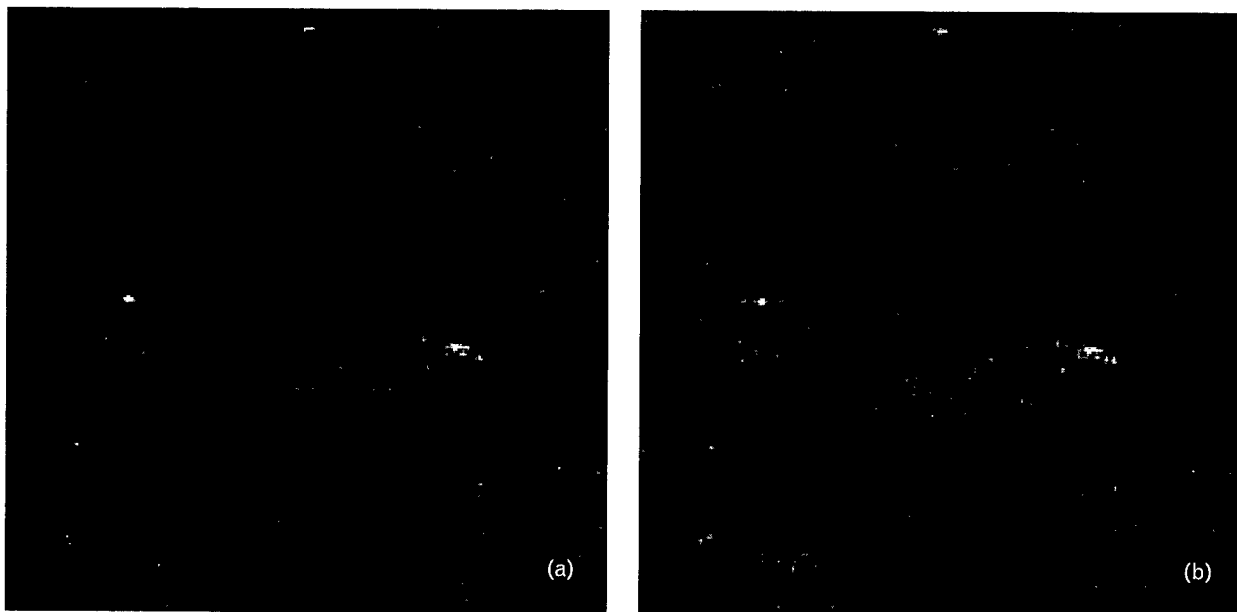
FIGURE 26. Adaptive-scalar-quantizer test images. (a) Orientation-selective bit allocation (0.53 bpp). (b) Isotropic bit allocation (0.54 bpp). Transform and quantization artifacts are clearly visible in these images.

NMXE, and the objectionable image has the highest NMXE.

Two types of image quality assessment experiments were conducted. In the first experiment, referred to as type A, nine images were displayed simultaneously with a reference image (identical to the original) in the center of the display. Subjects rated each image according to similarity to the reference image, on a scale of 0 to 4 with the following scale guide: 0 = very annoying differences (objectionable), 1 = annoying

FIGURE 27. Trellis-coded quantizer test images. (a) Orientation-selective bit allocation (0.54 bpp). (b) Isotropic bit allocation (0.50 bpp). These images are similar in image quality, and clearly superior to the adaptive-scalar-quantizer images in Figure 26.
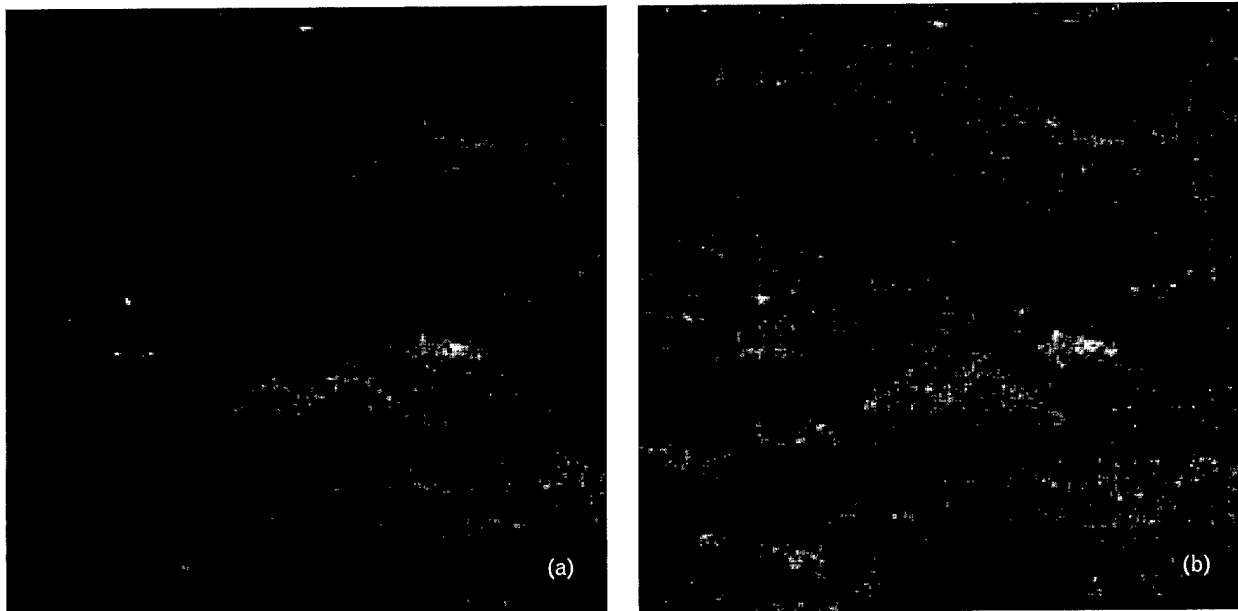
FIGURE 28. Vector-quantizer test images. (a) Block size is 1 × 1 for all frequency components (0.52 bpp). (b) Block sizes range from 1 × 1 for low-frequency components to 4 × 4 for high-frequency components (0.49 bpp). Although the texture quality is good in these images, the vector quantizer in this case adds undesirable false isolated point returns, which can be seen in the left center region of part *a* and in the right center region of part *b*.

differences, 2 = noticeable differences, 3 = just notice-able differences, and 4 = no differences (half-point ratings were allowed). In the second experiment, re-ferred to as type B, two images were displayed side by side and the reference was centered below the two test images. Subjects indicated which of the two test im-ages was most similar to the reference image by plac-ing the cursor inside the image region and clicking the left mouse button. If the subjects judged the two test images as equally similar or dissimilar to the refer-ence image, they were told to click on the reference image. After each selection, another pair of test im-

**Table 5. Bit Rates and Objective Quality Measures
for Each of the Test Images in Figures 25 through 29**

Image*	Bit Rate (bpp)	NMSE**	NMXE**
Objectionable	1.06	0.13549	0.64008
Adaptive scalar quantizer (OS)	0.53	0.13684	0.54345
Adaptive scalar quantizer (ISO)	0.54	0.13606	0.50816
Trellis-coded quantizer (OS)	0.54	0.13079	0.42353
Trellis-coded quantizer (ISO)	0.50	0.13449	0.51373
Vector quantizer (block size = 1)	0.52	0.13963	0.49662
Vector quantizer (1 ≤ block size ≤ 4)	0.49	0.15398	0.55832
Joint Photographic Experts Group (JPEG)	0.52	0.10669	0.47843

*   OS is orientation-selective bit allocation; ISO is isotropic bit allocation.
** NMSE is normalized mean-square error; NMXE is normalized maximum absolute error.

FIGURE 29. Joint Photographic Experts Group (JPEG) test image (0.52 bpp). Annoying block artifacts, characteristic of the block-based discrete cosine transform used in the JPEG image-compression standard, are apparent in this image.

ages was displayed and the cycle continued until all thirty-six image pairs were displayed.

The advantage of experiment type A is that all im-

ages can be viewed at once, and relative differences between each image and the reference image can be compared. Most subjects take advantage of the fact that all images are displayed at once and try to make their ratings reflect the relative differences. The disadvantage of experiment type A is that a comparison of images is more difficult because subjects have to make fairly large eye movements to compare all of the images, and subjects have to decide on a rating value for each different image (one of nine possible values).

The advantage of experiment type B is that fewer and smaller eye movements are required, and the decision process is much easier—there are only three choices. The disadvantage of experiment type B is that the subject has to view thirty-six image pairs for each trial, so the selection process tends to take longer, and some subjects report that the images tend to "fuzz over" after so many images are viewed. The results from experiment type B can be compared to those of experiment type A by assigning a score of +1 to "more similar" responses, −1 to the corresponding "less similar" responses, and 0 to the "equally dissimilar" responses. The total scores for each image are scaled to the interval [0,4].

The original and an "objectionable" image were in-

Table 6. Mean and Standard Deviation of Subjective Ratings for Each of the Test Images in Figures 25 through 29 *

Image	Experiment Type A		Experiment Type B	
	Mean	SD	Mean	SD
Original	3.9	0.3	4.0	0.0
Objectionable	0.1	0.5	0.2	0.3
Adaptive scalar quantizer (OS)	1.1	0.7	1.4	0.7
Adaptive scalar quantizer (ISO)	1.3	0.9	1.4	0.6
Trellis-coded quantizer (OS)	2.7	0.7	2.8	0.7
Trellis-coded quantizer (ISO)	2.7	0.8	2.7	0.7
Vector quantizer (block size = 1)	1.3	0.7	1.3	0.8
Vector quantizer (1 ≤ block size ≤ 4)	1.6	0.5	1.3	1.1
JPEG	1.5	0.7	1.5	1.1

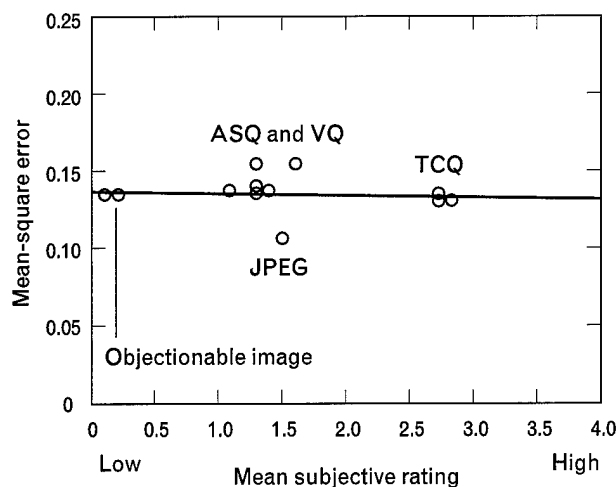* Based on responses from eleven subjects.

cluded in the set of test images for scaling purposes. The objectionable image was compressed by using ASQ with only two quantization levels for each of the modulated (AC) components. The objectionable image was characterized by noticeable salt-and-pepper noise and severe distortion of isolated point returns.

The displayed images were 256 × 256 subimage cutouts from the full size (512 × 512) test images. The experiments were conducted under dark ambient lighting conditions. The images were displayed on a Sony GDM-1961 monitor with a Digital Equipment Corporation Alpha computer system. The viewing distance was about eighteen inches and the test images subtended about 9.5° of visual angle. In both types of experiments, there were no constraints on viewing time.

The difficulty of experiment type A is reflected in the fact that some subjects were not able to identify the original as the most similar to the reference image. In spite of the difference in task difficulty, the correspondence between the two tests is remarkable. In fact, according to paired and unpaired Student's T tests for significance at the $p = 0.05$ level [51], there is no statistically significant difference between the corresponding means ($p > 0.21$ for the unpaired test and $p > 0.15$ for the paired test).

Table 6 summarizes the results of this experiment in tabular form, and Figure 30 summarizes these same results by plotting the mean subjective ratings as a function of the NMSE. The red line in Figure 30 represents a least-squares regressive linear fit. This figure clearly shows that the mean subjective ratings are essentially uncorrelated from the NMSE. From Table 6, the mean subjective ratings for both types of experiments reveal four distinct groupings: (1) the original image, (2) the objectionable image, (3) the TCQ images, and (4) the other images. Each of these groups has means that, according to paired and unpaired Student's T tests for significance at the $p = 0.05$ level, are significantly different. Also, the differences in means within these groups do not reflect statistically significant differences. Thus both TCQ images were rated as the most similar to the original, and they were superior to all other reconstructed images. There is no statistically significant difference between the mean ratings for ASQ and VQ and the JPEG image.

Summary

In this article we describe how general image-coding concepts relate to the compression of SAR imagery. We focus on the compression of detected SAR imagery for image archiving applications in which the end users are typically image analysts. The transform or representation scheme is a critical component of any image-compression system at low bit rates because it determines the structure of artifacts when very few transform coefficients are not quantized to zero.

We studied two complete transforms in detail: the Gabor transform and the Gabor-like wavelet packet (generalized wavelet) transform. At low bit rates, both of these transforms preserve isolated point returns well; however, each of these transforms has individual strengths and weaknesses. The Gabor transform tends to preserve textures well, and the Gabor-like wavelet packet transform tends to preserve textures and shadows well. Conventional wavelet decompositions tend to minimize mean-square error and preserve isolated-point-return quality well.

FIGURE 30. Normalized mean-square error as a function of mean subjective ratings averaged across eleven subjects. The red line represents a least-squares linear fit, and shows that mean-square error and mean subjective ratings are essentially uncorrelated. In this experiment, the system with the trellis-coded quantizer (TCQ) yielded higher mean subjective ratings than systems with the adaptive scalar quantizer (ASQ) or the vector quantizer (VQ). The TCQ system also had a significantly higher mean subjective rating than the JPEG image-compression standard, even though the JPEG standard had a lower mean-square error.

Our preferred compression system for the highest perceived SAR image quality consists of a wavelet packet transform that uses a Gabor-like tree structure with smooth biorthogonal wavelet filters. Currently, a universal trellis-coded quantizer followed by an arithmetic entropy coder seems to provide the best trade-off between complexity, accuracy, and bit rate. A distortion-minimizing and rate-minimizing bit-allocation procedure is used with the wavelet packet transform. The use of perceptually weighted distortion measures with the wavelet packet transform remains a topic for future investigation.

Other future directions of our research include developing better perceptual distortion metrics, adapting the decomposition to each test image, applying our system to applications in which algorithms are the end users (such as automatic target detection, automatic target recognition, and superresolution), designing distortion measures that are consistent with these applications, and applying general coding concepts to the compression of other data types such as SAR phase history data and multispectral data.

Acknowledgments

REFERENCES

1. D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE* **40**, Sept. 1952, pp. 1098–1101.
2. I.H. Witten, R.M. Neal, and J.G. Cleary, "Arithmetic Coding for Data Compression," *Commun. ACM* **30** (6), 1987, pp. 520–540.
3. A.M. Waxman, C. Lazott, D.A. Fay, A. Gove, and W.R. Steele, "Neural Processing of SAR Imagery for Enhanced Target Detection," *SPIE* **2487**, 1995, pp. 201–210.
4. G.K. Wallace, "The JPEG Still Picture Compression Standard," *Commun. ACM* **34** (4), 1991, pp. 30–44.
5. W.B. Pennebaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard* (Van Nostrand Reinhold, New York, 1993).
6. M. Porat and Y.Y. Zeevi, "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision," *IEEE Trans. Pattern Anal. Machine Intell.* **10** (4), 1988, pp. 452–468.
7. D. Gabor, "Theory of Communication," *J. Inst. Elec. Eng.* **93** (pt. III), 1946, pp. 429–457.
8. J.G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Opt. Soc. Am. A* **2** (7), 1985, pp. 1160–1169.
9. M. Bastiaans, "Gabor's Expansion of a Signal into Gaussian Elementary Signals," *Proc. IEEE* **68** (4), 1980, pp. 538–539.
10. J.G. Daugman, "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression," *IEEE Trans. Acoust. Speech Signal Process.* **36** (7), 1988, pp. 1169–1179.
11. L. Auslander, I.C. Gertner, and R. Tolimieri, "The Discrete Zak Transform Application to Time-Frequency, Analysis and Synthesis of Nonstationary Signals," *IEEE Trans. Signal Process.* **39** (4), 1991, pp. 825–835.
12. I. Gertner and Y.Y. Zeevi, "On the Zak-Gabor Representation of Images," *SPIE* **1360** (pt. 3), 1990, pp. 1738–1748.
13. K. Assaleh, Y.Y. Zeevi, and I. Gertner, "On the Realization of the Zak-Gabor Representation of Images," *SPIE* **1606** (pt. 1), 1991, pp. 532–540.
14. S.G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. Acoust. Speech Signal Process.* **37** (12), 1989, pp. 2091–2110.
15. S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Anal. Machine Intell.* **11** (7), 1989, pp. 674–693.
16. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. Image Process.* **1** (2), 1992, pp. 205–220.
17. R.R. Coifman and M.V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection," *IEEE Trans. Inf. Theory* **38** (2), 1992, pp. 713–718.
18. M.V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software,* chap. 8 (A.K. Peters, Wellesley, Mass., 1994).
19. K. Ramchandran and M. Vetterli, "Best Wavelet Packet Bases in a Rate-Distortion Sense," *IEEE Trans. Image Process.* **2** (2), 1993, pp. 160–175.
20. L.M. Novak, M.C. Burl, R.D. Chaney, and G.J. Owirka, "Optimal Processing of Polarimetric Synthetic-Aperture Radar Imagery," *Linc. Lab. J.* **3** (2), 1990, pp. 273–290.
21. S.P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans.*

Inf. Theory **28** (2), 1982, pp. 129–137.

22. J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Inf. Theory* **6** (1), 1960, pp. 7–12.

23. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-Constrained Vector Quantization," *IEEE Trans. Acoust. Speech Signal Process.* **37** (1), 1989, pp. 31–42.

24. R.L. Joshi, V.J. Crump, and T.R. Fischer, "Image Subband Coding Using Arithmetic Coded Trellis Coded Quantization," *IEEE Trans. Circuits Syst. Video Technol.* **5** (6), 1995, pp. 515–523.

25. C.E. Shannon, "A Mathematical Theory of Communication," pt. 1, *Bell System Tech. J.* 27 (3), 1948, pp. 379–423; "A Mathematical Theory of Communication," pt. 3, *Bell System Tech. J.* 27 (4), pp. 623–656.

26. A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression* (Kluwer Academic Publishers, Boston, 1992).

27. Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.* **28** (1), 1980, pp. 84–95.

28. R.M. Gray, "Time-Invariant Trellis Encoding of Ergodic Discrete-Time Sources with a Fidelity Criterion," *IEEE Trans. Inf. Theory* **23** (1), 1977, pp. 71–83.

29. E. Ayanoğlu and R.M. Gray, "The Design of Predictive Trellis Waveform Coders Using the Generalized Lloyd Algorithm," *IEEE Trans. Commun.* **34** (11), 1986, pp. 1073–1080.

30. G.D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE* **61** (3), 1973, pp. 268–278.

31. M.W. Marcellin, "Trellis Coded Quantization: An Efficient Technique for Data Compression," Ph.D. dissertation, Texas A&M University, 1987.

32. M.W. Marcellin and T.R. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," *IEEE Trans. Commun.* **38** (1), 1990, pp. 82–93.

33. T.R. Fischer and M. Wang, "Entropy-Constrained Trellis-Coded Quantization," *IEEE Trans. Inf. Theory* **38** (2), 1992, pp. 415–426.

34. M.W. Marcellin, "On Entropy-Constrained Trellis Coded Quantization," *IEEE Trans. Commun.* **42** (1), 1994, pp. 14–16.

35. J.H. Kasner, M.W. Marcellin, and B.R. Hunt, "Universal Trellis Coded Quantization," *IEEE Trans. Image Proc.,* preprint. 1997.

36. J.M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans. Signal Process.* **41** (12), 1993, pp. 3445–3463.

37. P. Sriram and M.W. Marcellin, "Image Coding Using Wavelet Transforms and Entropy-Constrained Trellis-Coded Quantization," *IEEE Trans. Image Process.* **4** (6), 1995, pp. 725–733.

38. A. Said and W.A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits Syst. Video Technol.* **6** (3), 1996, pp. 243–250.

39. Z. Xiong, K. Ramchandran, and M.T. Orchard, "Space-Frequency Quantization for Wavelet Image Coding," *IEEE Trans. Image Process.* **6** (5), 1997, pp. 677–693.

40. N.S. Jayant and P. Noll, *Digital Coding of Waveforms* (Prentice-Hall, Englewood Cliffs, N.J., 1984).

41. Y. Shoham and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Trans. Acoust. Speech Signal Process.* **36** (9), 1988, pp. 1445–1453.

42. D H. Kelly, "Effects of Sharp Edges on the Visibility of Sinusoidal Gratings," *J. Opt. Soc. Am.* **60** (1), 1970, pp. 98–103.

43. J.L. Mannos and D.J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," *IEEE Trans. Inf. Theory* **20** (4), 1974, pp. 525–536.

44. M.G. Perkins and T. Lookabaugh, "A Psychophysically Justified Bit Allocation Algorithm for Subband Image Coding," *Int. Conf. on Acoustics, Speech, and Signal Processing, Glasgow, Scotland, 23–26 May 1999,* pp. 1815–1818.

45. R.H. Bamberger and M.J.T. Smith, "A Comparison of Directionally Based and Non-Directionally Based Subband Image Coders," *SPIE* **1605** (pt. 2), 1991, pp. 757–768.

46. M.M. Taylor, "Visual Discrimination and Orientation," *J. Opt. Soc. Am.* **53** (6), 1963, pp. 763–765.

47. M.A. Webster and R.L. De Valois, "Relationship between Spatial-Frequency and Orientation Tuning of Striate-Cortex Cells," *J. Opt. Soc. Am. A* **2** (7), 1985, pp. 1124–1132.

48. D.R. Fuhrmann, J.A. Baro, and J.R. Cox, "Experimental Evaluation of Psychophysical Distortion Metrics for JPEG-Encoded Images," *SPIE* **1913,** 1993, pp. 179–190.

49. T.D. Penrod and G.G. Kuperman, "Image Quality Analysis of Compressed Synthetic Aperture Radar Imagery," Wright-Patterson AFB Armstrong Laboratory Technical Report AL/CF-TR-1993-0156, 1993.

50. N. Chaddha and T.H.Y. Meng, "Psycho-Visual Based Distortion Measure for Monochrome Image Compression," *SPIE* **2094** (pt. 3), 1993, pp. 1680–1690.

51. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing,* 2nd ed. (Cambridge University Press, Cambridge, U.K., 1992), pp. 615–618.

ROBERT BAXTER
was a staff member in the Machine Intelligence Technology group from 1994 to 1998. He has B.S. and M.S. degrees in electrical engineering from North Carolina State University and Ohio State University, respectively, and a Ph.D. degree in cognitive and neural systems from Boston University, where he received an award for excellence in teaching. His Ph.D. thesis developed a biologically and perceptually motivated approach to understanding and designing neural networks for tracking and recognizing objects in motion. Since 1993 he has been a visiting professor in the Neurological Surgery department at the University of Virginia, where he studies the function of the hippocampus and its relation to memory. At Lincoln Laboratory he worked on image-coding systems with an emphasis on how human analysts perceive and recognize features in synthetic aperture radar imagery, and he helped develop an image-compression system that was selected as the baseline for an international image-compression standard. He also worked on projects involving the perception and recognition of various types of acoustic signals. He has participated in several start-up companies, authored numerous technical papers as well as software-simulation tools and packages, and served as a consultant to several corporations. He is now with the Sensimetrics Corporation in Somerville, Massachusetts.

MICHAEL SEIBERT
is a staff member at the Lincoln Laboratory Kwajalein radar-measurements field site in the Marshall Islands, where he leads work on the KMR ID System (KIDS) for semi-automatic identification of reentry objects based on multisensor information fusion. He received B.S. and M.E. degrees in computer-systems engineering from Rensselaer Polytechnic Institute, and he received a Ph.D. in computer vision from Boston University, where he was a Presidential University Graduate Fellow. He joined Lincoln Laboratory in 1989 with assignments in 3-D object recognition, neural networks, seismic discrimination, multisensor data fusion, and synthetic aperture radar image processing. Before coming to the Laboratory, he was assistant professor of electrical engineering and computer science at the United States Air Force Academy, where he was nominated as outstanding military educator. He has also worked at Lawrence Livermore National Laboratory, and he has worked for General Electric at their Electronics Laboratory and Corporate Research and Development. He has written several book chapters, holds one patent, and has won several best-paper awards for his work. He is also on the editorial board of the journal *Neural Networks*.